



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Múltiplos Enxames Combinados com Métodos de Arquivamento, Pontos de Referência e Topologias na Otimização com Muitos Objetivos

Dissertação de Mestrado

José Lucas Matos Silva



São Cristóvão – Sergipe

2017

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

José Lucas Matos Silva

**Múltiplos Enxames Combinados com Métodos de
Arquivamento, Pontos de Referência e Topologias na
Otimização com Muitos Objetivos**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Prof. Dr. André Britto de Carvalho

São Cristóvão – Sergipe

2017

S586m SILVA, José Lucas Matos
Múltiplos Enxames Combinados com Métodos de Arquivamento, Pontos de Referência e Topologias na Otimização com Muitos Objetivos/ José Lucas Matos Silva. – São Cristóvão – Sergipe, 2017-
90 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. André Britto de Carvalho

Dissertação de Mestrado – UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, 2017.

1. Otimização com Muitos Objetivos. 2. Múltiplos Enxames. 3. Métodos de Arquivamento. 4. Pontos de Referência. I. Prof. Dr. André Britto de Carvalho. II. Múltiplos Enxames Combinados com Métodos de Arquivamento, Pontos de Referência e Topologias na Otimização com Muitos Objetivos.

CDU 004.021

José Lucas Matos Silva

**Múltiplos Enxames Combinados com Métodos de
Arquivamento, Pontos de Referência e Topologias na
Otimização com Muitos Objetivos**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Trabalho aprovado. São Cristóvão – Sergipe, 24 de Agosto de 2017:

Prof. Dr. André Britto de Carvalho
Orientador

Prof. Dr. Leonardo Nogueira Matos
Interno a instituição

Prof. Dr. Arion de Campos Junior
Externo a instituição

São Cristóvão – Sergipe
2017

Resumo

Problemas de Otimização Multiobjetivo podem ser classificados como um conjunto de problemas que possuem mais de uma função objetivo conflitante. Nestes problemas, as funções objetivo a serem otimizadas possuem índices de desempenho que são conflitantes, ou seja, normalmente quando um valor de uma função objetivo tem uma melhoria, um valor de outra função objetivo tende a ter uma piora. Com isto, é necessário obter um conjunto de melhores soluções, onde os valores das funções objetivo simultaneamente sejam aceitáveis. Pode-se destacar que nessa classe de problemas o número de melhores soluções cresce exponencialmente conforme o número de objetivos aumenta. Nesse sentido, esse aumento do número de soluções causa a deterioração da busca por melhores soluções, dificultando o progresso em direção às soluções ótimas. Apesar da aplicação com sucesso de diversos Algoritmos Evolucionários Multiobjetivo a esses tipos problemas, a maioria dos estudos foca em problemas com um número pequeno de objetivos. Além disso, esses algoritmos sofrem com a deterioração da busca quando o número de funções objetivo otimizadas é maior do que três. Assim, recentemente há a busca por novas técnicas e algoritmos que busquem reduzir a deterioração dos algoritmos multiobjetivo. A área que estuda essas novas técnicas é chamada de Otimização com Muitos Objetivos e os problemas multiobjetivo que possuem mais de três funções são classificados como Problemas de Otimização com Muitos Objetivos. Recentemente, diferentes abordagens têm sido propostas para melhorar o desempenho destes algoritmos em problemas de otimização com muitos objetivos. Uma dessas abordagens é o uso de múltiplas populações na otimização multiobjetivo por enxames de partículas, no qual chamamos de Múltiplos Enxames. Múltiplos Enxames são técnicas para explorar populações paralelas para decompor o problema e otimizá-lo de forma colaborativa. Neste trabalho foram desenvolvidos algoritmos que incorporam as características de múltiplos enxames com topologias, métodos de arquivamento e pontos de referência para resolver Problemas de Otimização com Muitos Objetivos. O primeiro algoritmo desenvolvido envolveu métodos de arquivamento e topologias, outro explorou pontos de referência e o último trabalhou com modificações dos pontos de referência com o objetivo de obter boa convergência e diversidade nesses tipos de problemas. Um conjunto de experimentos é feito para avaliar os algoritmos propostos e buscar identificar qual a melhor configuração de cada algoritmo. Além disso, é feita uma análise do desempenho dos algoritmos, comparando-os à métodos da literatura.

Palavras-chave: Otimização com Muitos Objetivos, Múltiplos Enxames, Métodos de Arquivamento, Pontos de Referência.

Abstract

Multi-Objective Optimization Problems can be classified as a set of problems that have more than one conflicting objective function. In these problems, the objective functions to be optimized have performance indexes that are conflicting, that is, usually when one value of an objective function has an improvement, a value of another objective function tends to worsen. With this, it is necessary to obtain a set of better solutions, where the values of the objective functions are simultaneously acceptable. It can be emphasized that in this class of problems the number of best solutions increases exponentially as the number of objectives increases. In this sense, this increase in the number of solutions causes a deterioration in the search for better solutions, making progress towards optimum solutions difficult. Despite the successful application of several Multiobjective Evolutionary Algorithms to these types of problems, most studies focus on problems with a small number of objectives. In addition, these algorithms suffer from search deterioration when the number of optimized objective functions is greater than three. Thus, recently there is the search for new techniques and algorithms that seek to reduce the deterioration of multiobjective algorithms. The area that studies these new techniques is called Many-Objective Optimization and multiobjective problems that have more than three functions are classified as Many-Objective Optimization Problems. Recently, different approaches have been proposed to improve the performance of these algorithms in optimization problems with many objectives. One of these approaches is the use of multiple populations in multi-objective particle swarm optimization, which we call Multiple Swarms. Multiple Swarms are techniques for exploring parallel populations to decompose the problem and optimize it in a collaborative way. In this work we developed algorithms that incorporate the characteristics of multiple swarms with topologies, archiving methods and reference points to solve Many-Objective Optimization Problems. The first algorithm developed involved archiving methods and topologies, another explored reference points, and the latter worked on modifications of reference points in order to achieve good convergence and diversity in these types of problems. A set of experiments is done to evaluate the proposed algorithms and seek to identify the best configuration of each algorithm. In addition, a performance analysis of the algorithms is performed comparing them to the literature methods.

Keywords: Many-Objective Optimization, Multi-Swarm, Archiving Methods, Reference Points.

Lista de ilustrações

Figura 1 – Imagem adaptada de (REYES-SIERRA; COELLO, 2006) que mostra a relação de dominância de Pareto.	17
Figura 2 – Fronteira de Pareto e o conjunto de soluções dominadas em um espaço com duas funções objetivo.	18
Figura 3 – Porcentagem de vetores não-dominados versus o número de objetivos. . . .	28
Figura 4 – Múltiplos enxames com diferentes topologias.	38
Figura 5 – Imagem de Oliveira e Carvalho (2017) que mostra um exemplo de hiperplano normalizado aplicado em 3 objetivos.	39
Figura 6 – Imagem de Oliveira e Carvalho (2017) que mostra o hiperplano em um formato côncavo.	41
Figura 7 – Imagem de Oliveira e Carvalho (2017) que mostra o hiperplano em um formato convexo.	42
Figura 8 – Imagem de Oliveira e Carvalho (2017) que mostra o hiperplano em um formato senoidal.	42
Figura 9 – <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2.	49
Figura 10 – <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.	50
Figura 11 – <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ4.	51
Figura 12 – <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ6.	52
Figura 13 – Valores médios do indicador de qualidade GD para o problema DTLZ2. . .	55
Figura 14 – Valores médios do indicador de qualidade IGD para o problema DTLZ2. . .	55
Figura 15 – Valores médios do indicador de qualidade GD para o problema DTLZ4. . .	56
Figura 16 – Valores médios do indicador de qualidade IGD para o problema DTLZ4. . .	56
Figura 17 – Valores médios do indicador de qualidade GD para o problema DTLZ6. . .	57
Figura 18 – Valores médios do indicador de qualidade IGD para o problema DTLZ6. . .	57
Figura 19 – Valores médios do indicador de qualidade GD para o problema DTLZ2 Convexo.	58
Figura 20 – Valores médios do indicador de qualidade IGD para o problema DTLZ2 Convexo.	58
Figura 21 – Quantidade de pontos de referência: <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2. . .	61

Figura 22 – Quantidade de pontos de referência: <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.	62
Figura 23 – Quantidade de pontos de referência: <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ4. . .	63
Figura 24 – Quantidade de pontos de referência: <i>Boxplot</i> gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ6. . .	64
Figura 25 – Função côncava: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.	68
Figura 26 – Função côncava: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.	69
Figura 27 – Função côncava: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4	70
Figura 28 – Função côncava: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.	71
Figura 29 – Função convexa: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.	73
Figura 30 – Função convexa: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.	74
Figura 31 – Função convexa: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4	75
Figura 32 – Função convexa: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.	76
Figura 33 – Função senoidal: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.	78
Figura 34 – Função senoidal: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.	79
Figura 35 – Função senoidal: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4	80
Figura 36 – Função senoidal: <i>Boxplot</i> dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.	81

Lista de tabelas

Tabela 1 – Parâmetros do NSGA-III	46
Tabela 2 – Arquivadores	47
Tabela 3 – Combinações entre os arquivadores	47
Tabela 4 – Análise resumida das melhores combinações de arquivadores	48
Tabela 5 – Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)	54
Tabela 6 – Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)	54
Tabela 7 – Quantidade de pontos de referência: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)	60
Tabela 8 – Quantidade de pontos de referência: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)	60
Tabela 9 – Comparação MSAT, MSAT-H e NSGA-III para o indicador GD.	65
Tabela 10 – Comparação MSAT, MSAT-H e NSGA-III para o indicador IGD.	65
Tabela 11 – Função côncava: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)	67
Tabela 12 – Função côncava: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)	67
Tabela 13 – Função convexa: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)	72
Tabela 14 – Função convexa: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)	72
Tabela 15 – Função senoidal: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)	77
Tabela 16 – Função senoidal: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)	77
Tabela 17 – Comparação MSAT, MSAT-H, MSAT-MH e NSGA-III para o indicador GD.	82
Tabela 18 – Comparação MSAT, MSAT-H, MSAT-MH e NSGA-III para o indicador IGD.	82

Lista de códigos

Algoritmo 1 – Pseudocódigo do algoritmo PSO	22
Algoritmo 2 – Pseudocódigo do algoritmo MOPSO	24
Algoritmo 3 – Algoritmo MSAT	37
Algoritmo 4 – Algoritmo MSAT-H	40
Algoritmo 5 – Algoritmo MSAT-MH	43

Lista de abreviaturas e siglas

GD	<i>Generational Distance</i> (Distância Geracional)
IGD	<i>Inverted Generational Distance</i> (Distância Geracional Invertida)
MaOPs	<i>Many-Objective Optimization Problems</i> (Problemas de Otimização com Muitos Objetivos)
MGA	<i>Multi-Level Grid Archiving</i>
MOEA	<i>Multi-Objective Evolutionary Optimization Algorithms</i> (Algoritmos Evolucionários Multiobjetivo)
MOPs	<i>Multi-Objective Optimization Problems</i> (Problemas de Otimização Multi-Objetivo)
MOPSO	<i>Multiobjective Particle Swarm Optimization</i> (Otimização Multiobjetivo por Enxame de Partículas)
MSAT	Algoritmo Multi-Swarm baseado em Arquivamento e Topologias
NSGAI	<i>Non-dominated Sorting Genetic Algorithm II</i>
PF	<i>Pareto Front</i> (Fronteira de Pareto)
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)

Sumário

1	Introdução	12
2	Fundamentação Teórica	15
2.1	Conceitos básicos da Otimização Multiobjetivo	15
2.1.1	Trabalhos relacionados aos MOPs	20
2.2	Otimização com Muitos Objetivos	27
2.2.1	Como resolver	28
2.2.2	Trabalhos relacionados aos MaOPs	29
2.3	Conclusões	33
3	Algoritmos de Múltiplos Enxames Aplicados a MaOPs	35
3.1	Algoritmo MSAT	36
3.2	Hiperplano	38
3.3	Modificações no Hiperplano	40
4	Experimentos	44
4.1	Metodologia	44
4.2	Problemas	45
4.3	Indicadores de Qualidade	45
4.4	Algoritmos e Parâmetros	46
4.4.1	Parâmetros Gerais	46
4.4.2	Parâmetros do Experimento I	46
4.4.3	Parâmetros do Experimento II	48
4.4.4	Parâmetros do Experimento III	48
4.5	Experimento I	48
4.5.1	Análise de parâmetros do MSAT	48
4.5.2	Resultados e Discussão	53
4.6	Experimento II	59
4.6.1	Análise de parâmetros do MSAT-H	59
4.6.2	Resultados e Discussão	65
4.7	Experimento III	66
4.7.1	Análise de parâmetros do MSAT-MH	66
4.7.2	Resultados e Discussão	82
5	Considerações Finais	84
	Referências	86

1

Introdução

Os Problemas de Otimização Multi-Objetivo (MOP, do inglês *Multi-Objective Optimization Problems*) são problemas que possuem mais de uma função objetivo conflitante. Com isto, é necessário obter um conjunto de melhores soluções, onde os valores das funções objetivo simultaneamente sejam aceitáveis e que auxilie no processo de tomada a decisão. A Teoria da Otimalidade de Pareto é utilizada para definir a Otimização Multi-Objetivo (COELLO; LAMONT; VELDHUIZEN, 2007).

Problemas de Otimização Multi-Objetivo são resolvidos por diferentes áreas de pesquisa e no geral são problemas da classe de complexidade NP-Difícil (COELLO; LAMONT; VELDHUIZEN, 2007). Devido a essa complexidade, os MOPs têm impossibilitado a utilização de algoritmos de busca tradicionais. Dessa forma, o uso de Algoritmos Evolucionários Multiobjetivo (MOEA, do inglês *Multi-Objective Evolutionary Optimization Algorithms*) (COELLO; LAMONT; VELDHUIZEN, 2007) e de outras meta-heurísticas tornaram-se comum por terem resultados satisfatórios em termos de convergência e diversidade. Exemplos de algoritmos que compõem essas meta-heurísticas estão os algoritmos baseados em *Swarm Intelligence* (como Otimização por Nuvem de Partículas (REYES-SIERRA; COELLO, 2006)) (POLI; KENNEDY; BLACKWELL, 2007), Algoritmos Genéticos (GOLDBERG; HOLLAND, 1988), entre outros.

Um algoritmo que apresenta bons resultados em trabalhos na literatura com a abordagem em MOPs é o algoritmo MOPSO (do inglês, *Multi-Objective Particle Swarm Optimization*) (REYES-SIERRA; COELLO, 2006). Esta meta-heurística tem como base o mesmo conceito do algoritmo PSO (do inglês, *Particle Swarm Optimization*), onde é classificado como um algoritmo de inteligência de enxames e que surgiu da análise comportamental de vários pássaros que visavam encontrar comida. No PSO, um indivíduo é chamado de partícula, onde cada partícula segue simples regras que vão controlar as suas interações com o restante da população, fazendo dessas interações um comportamento inteligente (KENNEDY; EBERHART, 1995a).

Apesar de bons resultados, as abordagens discutidas nos parágrafos anteriores deterioram quando o número de funções objetivo cresce. Problemas de Otimização com Muitos Objetivos (do inglês, *Many-Objective Optimization Problems* ou MaOPs) são problemas que têm mais de três funções objetivo e a área que tenta resolver esses tipos de problemas é chamada de Otimização com Muitos Objetivos.

Vários trabalhos na literatura tentam resolver problemas da categoria dos MaOPs, como exemplo do NSGA-III que usa pontos de referência (DEB; JAIN, 2014a). Dentre esses trabalhos, também existem os que se destacam no uso da técnica de arquivamento para controlar a convergência e diversidade, baseado em múltiplos enxames. Um algoritmo que é referência na abordagem de múltiplos enxames é o MOPSO. Além dele, existe os trabalhos de Britto, Mostaghim e Pozo (2014) e Campos, Pozo e Duarte (2013) que utilizam a estratégia de múltiplos enxames com arquivamento.

Apesar de muitos trabalhos possuírem bons resultados em termos de convergência e diversidade, existem algumas limitações que devem ser destacadas. Essas limitações devem ser consideradas e trabalhadas para levar a uma melhoria do funcionamento do algoritmo no uso de problemas com muitos objetivos. Essas limitações foram extraídas de trabalhos relacionados que utilizaram diferentes abordagens para resolução dos MaOPs. Alguns exemplos de limitações são: a utilização de parâmetros constantes para controlar o grau de expansão e compressão da área de dominância das soluções; não execução de testes em diferentes *benchmarks* para comprovar com mais lealdade os resultados; não exploração de técnicas sobre topologia aplicada a muitos objetivos; o uso fixo dos parâmetros de configuração; não trabalhar de forma adequada com pontos de referência, no qual os pontos geralmente utilizados são fixos.

De acordo com os resultados obtidos em (CAMPOS; POZO; DUARTE, 2013; BRITTO; MOSTAGHIM; POZO, 2014; BRITTO; POZO, 2012b; BRITTO; POZO, 2014; BRITTO; POZO, 2015) apesar das limitações, a abordagem de múltiplos enxames, arquivos e pontos de referência é um bom caminho a se seguir e os algoritmos baseados no MOPSO se destacam devido aos melhores resultados encontrados quando comparados à outras meta-heurísticas que utilizam a abordagem de problemas com muitos objetivos. Dessa forma, o seguinte trabalho visa investigar uma maneira de fazer uma combinação entre múltiplos enxames, métodos de arquivamento e pontos de referência.

Esta dissertação de mestrado tem como objetivo desenvolver técnicas de múltiplos enxames, combinada com métodos de arquivamento e pontos de referência para resolver problemas de otimização com muitos objetivos. Além de ter objetivos específicos como: desenvolver topologias como padrões de comunicação; desenvolver pontos de referência para guiar as soluções para o espaço onde possivelmente teriam pontos ideais; investigar a qualidade dos algoritmos em cenários com muitos objetivos.

Como metodologia primeiro foi necessário fazer um estudo do estado da arte e das técnicas para implementar um MOPSO com múltiplos enxames. A seguir, foi implementado

a abordagem de diferentes topologias e associado diferentes métodos de arquivamento. Depois, foram propostos dois novos algoritmos de múltiplos enxames que buscam combinar as novas topologias aos pontos de referência. E com intuito de validar os algoritmos propostos, o desempenho foi avaliado em problemas de *benchmark*. Os algoritmos desenvolvidos foram avaliados utilizando uma metodologia que explore problemas com muitos objetivos. Para isso foram utilizados os problemas da família DTLZ, disponíveis no *framework* jMetal (DURILLO; NEBRO, 2011). Os métodos foram avaliados em termos de convergência e diversidade utilizando indicadores de qualidade como o GD e IGD (CARVALHO, 2013). Os algoritmos foram escalados variando o número de objetivos entre 3 e 10 funções objetivos e para a comparação, foram utilizados algoritmos estado-da-arte da literatura.

A dissertação está organizada da seguinte forma: no Capítulo 2 são apresentados os conceitos básicos da Otimização Multiobjetivo e Otimização com Muitos Objetivos, os trabalhos relacionados a esses conceitos, e além disso são apresentados as limitações dos trabalhos da literatura; no capítulo 3 são apresentados os algoritmos desenvolvidos; no capítulo 4 são apresentados os experimentos juntamente com a metodologia experimental, além de mostrados os problemas que fizeram parte dos experimentos e quais indicadores de qualidade e parâmetros foram utilizados para análise dos algoritmos; e por último, são apresentadas algumas considerações finais no Capítulo 5.

2

Fundamentação Teórica

No capítulo corrente são apresentadas as definições da Otimização com Muitos Objetivos. São apresentados também os conceitos básicos da Otimização Multiobjetivo, bem como trabalhos relacionados relevantes da área. Com isso, é feita uma análise, mostrando quais foram as contribuições que esses trabalhos trouxeram e identificando os desafios que serão colocados em pauta nessa dissertação. A organização do capítulo está da seguinte forma: 2.1 apresenta a definição de otimização multiobjetivo; os trabalhos relacionados são apresentados na seção 2.1.1; na seção 2.2 é destacado o comportamento da otimização quando aplicado a problemas com mais de 3 objetivos; já na seção 2.2.1 é elencada como resolver as dificuldades encontradas na seção anterior; Por fim, a seção 2.2.2 exibe os trabalhos correlatos que tentam resolver as dificuldades encontradas na otimização multiobjetivo, aplicadas em problemas com mais de 3 objetivos.

2.1 Conceitos básicos da Otimização Multiobjetivo

A Otimização Multiobjetivo é aplicada em problemas com finalidade de otimizar simultaneamente várias funções objetivo, de forma que possa encontrar um conjunto de soluções onde representem com eficácia a relação entre os objetivos. Os Problemas de Otimização Multi-Objetivo (MOP, do inglês *Multi-Objective Optimization Problems*) podem ser classificados como um conjunto de problemas que possuem mais de uma função objetivo conflitante. As funções objetivo possuem índices de desempenho que são conflitantes, ou seja, normalmente quando um valor de uma função objetivo tem uma melhoria, um valor de outra função objetivo tende a ter uma piora. Com isto, é necessário obter um conjunto de melhores soluções, onde os valores das funções objetivo simultaneamente sejam aceitáveis (melhor valor possível em todos os objetivos) e que auxilie no processo de tomada a decisão. Com a necessidade de obter soluções simultaneamente aceitáveis, surgiu a Otimização Multiobjetivo, no qual apresenta uma definição teórica de problemas de otimização multiobjetivo, bem como um conjunto de algoritmos e técnicas para

resolver esses problemas (COELLO; LAMONT; VELDHUIZEN, 2007; CARVALHO, 2013).

Num problema de otimização uma solução pode ser entendida como um vetor que possua variáveis de decisão que atendem determinadas restrições. Para medir a qualidade de uma solução em um problema de otimização utiliza-se alguns critérios, nos quais são mostrados através de funções computáveis das variáveis de decisão. Em um problema multiobjetivo há várias funções a serem otimizadas, nas quais essas são ditas como funções objetivo e expressadas por $f_j(\vec{x})$, $F : R^n \rightarrow R$. Em problemas multiobjetivo existem várias funções objetivo, dessa forma as funções são inseridas em um vetor e esse passa a ser denotado por $\vec{f}(\vec{x}) \in \Lambda$, em que Λ é o espaço do vetor das funções objetivo para o determinado problema. Logo adiante é definida a expressão de otimização multiobjetivo (para um problema de minimização sem restrições):

$$\text{Minimiza } \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})) \quad (2.1)$$

em que o número de funções objetivo e a dimensão de Λ é denotado por m . Cada solução \vec{x} no espaço de variáveis de decisão pode ser mapeada para o vetor $\vec{f}(\vec{x})$ que pertence ao espaço de funções objetivo $F : R^n \rightarrow \Lambda$.

Os MOPs têm como meta a otimização de m funções objetivo em simultaneidade. Em termos matemáticos, o MOP é definido como minimização de $\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}))$ podendo $g_i(\vec{x}) \leq 0, i = \{1, \dots, m\}$, e $h_j(\vec{x}) = 0, j = \{1, \dots, p\}$ $\vec{x} \in \Omega$ (onde Ω é definido como um universo qualquer). Uma solução de MOP minimiza (ou maximiza) os elementos de um vetor $\vec{f}(\vec{x})$ em que \vec{x} é definido como uma variável de decisão com n dimensões $\vec{x} = (x_1, \dots, x_n) \in \Omega$. Nesse texto, por simplicidade, vamos adotar como padrão problemas de minimização.

A Otimização Multiobjetivo é definida através da Teoria da Otimalidade de Pareto. A solução ótima em problemas com muitos objetivos é definida de acordo com algumas teorias como a Dominância de Pareto, Pareto ótimo, Conjunto Pareto ótimo, Fronteira de Pareto, Vetor Ideal e Vetores Extremos (COELLO; LAMONT; VELDHUIZEN, 2007; CARVALHO, 2013). As definições de cada um destes termos dessas teorias são descritas logo a seguir:

Definição 1

Dominância de Pareto: Seja $\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}))$ e $\vec{f}(\vec{y}) = (f_1(\vec{y}), f_2(\vec{y}), \dots, f_m(\vec{y}))$, $\vec{f}(\vec{x})$ **domina** $\vec{f}(\vec{y})$, é expressado por $\vec{f}(\vec{x}) \preceq \vec{f}(\vec{y})$, unicamente se (minimização): $\forall i \in \{1, 2, \dots, m\} : f_i(\vec{x}) \leq f_i(\vec{y})$, e $\exists i \in \{1, 2, \dots, m\} : f_i(\vec{x}) < f_i(\vec{y})$, $\vec{f}(\vec{x})$ é dito não dominado caso não exista nenhum $\vec{f}(\vec{y})$ que tenha domínio sobre $\vec{f}(\vec{x})$.

Para melhor entender o conceito de dominância de Pareto num caso particular com um espaço de duas funções objetivos, é mostrado um exemplo na figura 1. Os círculos de cor rosa são soluções dominadas, porque existe pelo menos um $\vec{f}(\vec{y})$ que domina $\vec{f}(\vec{x})$ em pelo menos um dos objetivos, enquanto que o círculo de cor azul é uma solução não dominada porque não existe um $\vec{f}(\vec{y})$ que domina em pelo menos um dos objetivos.

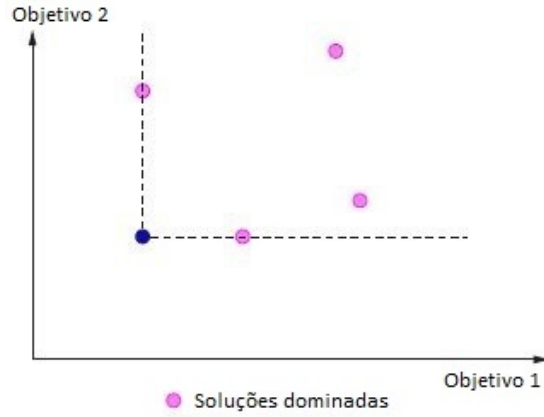


Figura 1 – Imagem adaptada de (REYES-SIERRA; COELLO, 2006) que mostra a relação de dominância de Pareto.

Definição 2

Pareto ótimo: Diz que um vetor \vec{x} é Pareto ótimo se seu vetor objetivo é não dominado. Um vetor solução $\vec{x} \in \Omega$, com suas funções objetivo $\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}))$, \vec{x} diz que **Pareto Ótimo** se unicamente não existe $\vec{y} \in \Omega$, com o vetor que representa as funções objetivo $\vec{f}(\vec{y}) = (f_1(\vec{y}), f_2(\vec{y}), \dots, f_m(\vec{y}))$, de forma que $\vec{f}(\vec{y}) \preceq \vec{f}(\vec{x})$. Como observado, é dito Pareto ótimo quando as funções objetivo do vetor solução não pode ser simultaneamente melhorada.

Definição 3

Conjunto Pareto ótimo: Num problema de otimização multiobjetivo, é demonstrado que um Conjunto Pareto ótimo P^* , é um conjunto que possui as melhores soluções em Ω , logo, as soluções Pareto ótimo definem o Conjunto Pareto ótimo do problema. O conjunto é denotado por:

$$P^* := \{\vec{x} \in \Omega \mid \nexists \vec{y} \in \Omega, \vec{f}(\vec{y}) \preceq \vec{f}(\vec{x})\} \quad (2.2)$$

Definição 4

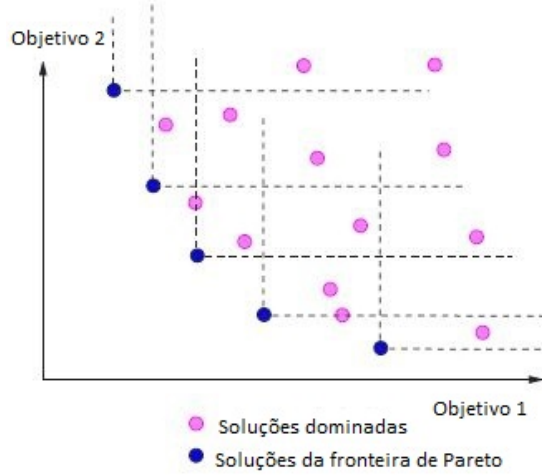
Fronteira de Pareto: Os vetores que não são dominados no espaço de funções objetivo acabam formando a Fronteira de Pareto (PF, do inglês *Pareto Front*), onde toda solução que fazer parte do Conjunto Pareto ótimo forma uma possível solução em um ponto não dominado no espaço de funções objetivo. A Fronteira de Pareto é definida por:

$$PF := \{\vec{f}(\vec{x}) \mid \vec{x} \in P^*\} \quad (2.3)$$

e pode ser mais entendível na forma de imagem, como mostra a figura 2. Os vetores dominados são representados pelos círculos de cor rosa, pois existe pelo menos um $\vec{f}(\vec{y})$ que domina $\vec{f}(\vec{x})$ em pelo menos um dos objetivos, enquanto que os círculos de cor azul são soluções não dominadas porque não existe um $\vec{f}(\vec{y})$ que domina em pelo menos um dos objetivos, logo, o conjunto de círculos de cor azul formam a PF do problema. Podemos

definir PF sendo como PF_{real} , onde é a fronteira de Pareto do problema e PF_{aprox} no qual é um conjunto de aproximação no espaço das funções objetivo.

Figura 2 – Fronteira de Pareto e o conjunto de soluções dominadas em um espaço com duas funções objetivo.



Fonte: adaptada pelo autor

Definição 5

Vetor Ideal: Seja um vetor de variáveis \vec{x} que otimiza a função objetivo $f(\vec{x})$, é dito que um vetor é ideal se possui os melhores valores para cada função objetivo. A definição do Vetor Ideal é expressada por:

$$\vec{f^0}(\vec{x}) = [f_1^0(\vec{x}), f_2^0(\vec{x}), \dots, f_m^0(\vec{x})] \quad (2.4)$$

no qual $f_i(\vec{x})$ é a função objetivo e $f_i^0(\vec{x})$ é o valor ótimo dela. Já o $\vec{f^0}(\vec{x})$ é um vetor utópico que é usado para poder definir a melhor posição no espaço de objetivos.

Definição 6

Vetores Extremos: Considere que $f_i^0(\vec{x})$ é o menor valor para um objetivo e $f_j^{max}(\vec{x})$ é o maior valor. Eles são vetores não dominados e que pertencem ao espaço de funções objetivo. Portanto, $f_i^0(\vec{x})$ e $f_j^{max}(\vec{x})$ são valores extremos e não dominados se cada um pertence a um ponto no eixo de objetivos, caso contrário, eles não são extremos. A elucidação de Vetores Extremos é mostrado a seguir:

$$\vec{E_j} = [f_1^0(\vec{x}), f_2^0(\vec{x}), \dots, f_j^{max}(\vec{x}), \dots, f_m^0(\vec{x})] \quad (2.5)$$

no qual $f_i(\vec{x})$ é a função objetivo e $f_i^0(\vec{x})$ é um valor ótimo. Já $f_j(\vec{x})$ é a função que está associada ao valor ótimo $f_j^{max}(\vec{x})$. Os m vetores geralmente são usados individualmente como representação de cada ponto em cada eixo de objetivos. O conjunto desses pontos

é denotado por \vec{E}_j . É importante ressaltar que o vetor extremo não corresponde a uma solução real, mas é criado para representar cada extremidade da fronteira de Pareto.

Os Problemas de Otimização Multiobjetivo são resolvidos por diferentes áreas de pesquisa e no geral são problemas da classe de complexidade NP-Difícil (COELLO; LAMONT; VELDHUIZEN, 2007). Devido a grande complexidade, os MOPs têm impossibilitado a utilização de algoritmos de busca tradicionais. Dessa forma, o uso de Algoritmos Evolucionários¹ e de outras meta-heurísticas tornaram-se comum por terem resultados satisfatórios. Exemplos de algoritmos que compõem essas meta-heurísticas estão os algoritmos baseados em *Swarm Intelligence*², Algoritmos Genéticos (GOLDBERG; HOLLAND, 1988), Otimização por Nuvem de Partículas (REYES-SIERRA; COELLO, 2006), entre outros. Geralmente, possuindo em comum o uso de uma população para a procura por soluções próximas a fronteira de Pareto, sendo possível encontrá-las em apenas uma execução. As técnicas de otimização baseadas no comportamento de seres vivos e que abordam mecanismos de busca incorporando conceitos multiobjetivo é chamada de Algoritmos Evolucionários Multiobjetivo (MOEA, do inglês *Multi-Objective Evolutionary Optimization Algorithms*).

Essa incorporação de conceitos multiobjetivo nos Algoritmos Evolucionários tem origem através das relações de preferências. Essas relações são determinantes para distinguir soluções boas de ruins. Nos MOEAs, existe uma função que é capaz de determinar se uma solução é boa ou ruim, o nome dessa função é chamada de *fitness*. A função *fitness* impõe as relações de preferência entre as soluções de uma população e isso ajuda a identificar regiões do espaço de busca com um melhor potencial (JAIMES; COELLO, 2009b). Logo, o *fitness* é responsável por definir as relações de preferência e determinar se uma solução \vec{x} é preferível a uma solução \vec{y} .

As abordagens de otimização multiobjetivo relacionadas a função *fitness* têm como característica marcante encontrar os elementos que compõe o conjunto Pareto ótimo. Ressaltando que essas meta-heurísticas não garantem o ótimo e tentam encontrar um conjunto de soluções o mais próximo do conjunto Pareto ótimo. O conjunto de soluções encontradas por esses algoritmos é conhecido como conjunto aproximado e sua imagem no espaço de objetivos, é chamada de fronteira aproximada ou (PF_{approx}). Para encontrar os elementos desse conjunto, geralmente, a principal relação de preferência adotada é o mecanismo da dominância de Pareto. Nos MOEAs é necessário mecanismos para ordenar as soluções. Assim, é necessário adaptar os conceitos para que seja possível definir essa ordenação. Esse conceito de adaptação de mecanismos de ordenação será melhor explorado na seção 2.2.

¹ Algoritmos Evolucionários são técnicas de otimização que normalmente são inspiradas no comportamento social e no processo de evolução de seres vivos (COELLO; LAMONT; VELDHUIZEN, 2007)

² É um conceito empregado a uma população de agentes que interagem localmente com outros e com o seu ambiente. Os agentes seguem simples regras e as interações entre eles faz nascer um comportamento global inteligente (POLI; KENNEDY; BLACKWELL, 2007)

2.1.1 Trabalhos relacionados aos MOPs

Nessa seção serão discutidos alguns trabalhos relacionados existentes na literatura que tem como foco a Otimização Multiobjetivo. Esses trabalhos que serão apresentados têm como objetivo destacar pontos importantes como os tipos de algoritmos multiobjetivo e suas evoluções.

Um dos trabalhos que aborda conceitos de otimização multiobjetivo é o trabalho apresentado por [Deb et al. \(2000\)](#). Nele é mostrado as facetas da construção do algoritmo NSGA e a sua evolução para o NSGA-II. O NSGA (do inglês *Non-Dominated Sorting Genetic Algorithm*) é um algoritmo evolucionário multiobjetivo que utiliza métodos de ordenação de soluções não-dominadas e que sua complexidade computacional é de $O(MN^3)$, onde M é o número total de objetivos e N é o tamanho da população. Além do NSGA ter uma complexidade computacional de $O(MN^3)$, ele é um algoritmo não-elitista e que utiliza um parâmetro de compartilhamento, o que pode causar alguns problemas:

- i. Com a complexidade computacional sendo $O(MN^3)$, o custo dessa ordenação pode ser muito alto quando a população for muito grande, principalmente quando a população for ordenada em cada geração;
- ii. Resultados ([DEB et al., 2000](#); [ZITZLER; DEB; THIELE, 2000](#); [RUDOLPH, 2001](#)) mostram que o elitismo é viável para acelerar o desempenho e reduzir a perda de boas soluções uma vez que já foram encontradas;
- iii. O parâmetro de compartilhamento é um mecanismo tradicional que tem como afimco criar diversidade, evitando assim a convergência prematura para algum ponto do espaço de busca, mas a sua especificação pode ser um problema.

Devido aos problemas descritos, foi criado o algoritmo NSGA-II que é uma versão melhorada do NSGA. O NSGA-II tem como finalidade tornar mais rápida a ordenação do que o algoritmo inicial, o NSGA. No NSGA-II, a fim de ordenar uma população com tamanho N com relação as soluções não-dominadas, cada solução é comparada a todas as outras soluções para verificar se ela é dominada. Essa metodologia de ordenação requer $O(MN)$ comparações para cada uma solução, onde M é notado como o número de objetivos. Com a busca para achar os membros não-dominados que vão compor a primeira fronteira, a complexidade computacional é de $O(MN^2)$. O método desconsidera a procura na primeira frente que já foi determinada como não-dominada, assim esse procedimento é repetido outras vezes para achar as fronteiras subsequentes, onde no pior caso a complexidade pode chegar a $O(MN^3)$ quando apenas existir uma solução para cada fronteira.

Uma das características que é admirada no NSGA-II é o elitismo. Essa característica é responsável por manter possíveis boas soluções, o que acaba otimizando a busca. Além do elitismo, outra característica admirável é a preservação da diversidade, no qual, é mantida por

um estimador de densidade chamado de distância de *Crowding* (do inglês, *Crowding Distance*). No trabalho de (DEB et al., 2000), o NSGA-II foi melhor quando comparado com outro MOEA (o algoritmo chamado PAES (KNOWLES; CORNE, 1999)) que tem como característica a preservação de soluções não-dominadas e que também é um algoritmo elitista. Assim como esse trabalho, outros trabalhos utilizam o NSGA-II como base de comparação por ter sucesso na obtenção dos resultados em vários MOPs.

Outro algoritmo que apresenta bons resultados em trabalhos na literatura é o algoritmo MOPSO. Esta meta-heurística tem como base o mesmo conceito do algoritmo PSO (do inglês, *Particle Swarm Optimization*), onde é classificado como um algoritmo evolutivo e que surgiu da análise comportamental de vários pássaros que visavam encontrar comida. No PSO, um indivíduo é chamado de partícula, onde cada partícula segue simples regras que vão controlar as suas interações com o restante da população, fazendo dessas interações um comportamento inteligente. Ainda no conceito do PSO, existem algumas terminologias que definem esse algoritmo (REYES-SIERRA; COELLO, 2006):

Swarm ou Enxame: é a população do algoritmo;

Partícula: Indivíduo do enxame. Sendo que cada partícula representa uma possível solução do problema, possuindo elementos de velocidade e posição;

***pbest*:** melhor posição que a partícula chegou até o momento, também chamada de líder local, ou seja, representa o componente local;

***lbest*:** melhor posição que a partícula membro de uma vizinhança chegou até momento;

***gbest*:** posição da melhor partícula de todo o enxame;

Líder: São partículas usadas para guiar uma outra partícula para uma melhor região no espaço de busca;

Velocidade (vetor): esse é o vetor de velocidade responsável por determinar para qual direção uma partícula deve movimentar, melhorando assim a atual posição;

Peso de inércia: o peso é usado para ter o controle da influência dos valores anteriores da velocidade no cálculo da velocidade corrente de uma dada partícula;

Fator de aprendizado: descreve como uma partícula será atraída para a direção de seu próprio sucesso (*pbest*), ou para o sucesso do enxame (*gbest*). Existe dois tipos de fatores de aprendizado e são definidos como cognitivo e social, onde o cognitivo representa a atração com que a partícula tem com seu próprio sucesso. Já o social, é responsável por representar a atração com que a partícula tem com o enxame;

Topologia de vizinhança: representa o conjunto de partículas que contribuem para o cálculo do *lbest* de cada partícula. As principais topologias usadas são: Grafo vazio; Estrela; Arvore; Grafo completo e Melhor local (CARVALHO, 2013).

Algoritmo 1 Pseudocódigo do algoritmo PSO

```

1: Inicializa o enxame
2: Localiza o líder
3: g = 0
4: while g < gmax do
5:   for Cada Partícula do
6:     Atualiza Posição(Vôo)
7:     Avaliação
8:     Atualiza o pbest
9:   Atualiza o líder
10:  g++

```

Antes de falar do pseudocódigo, é importante explicar que a atualização da posição da partícula é definida por equações de movimento e essas equações guiarão a partícula para um espaço que contenha as melhores soluções do problema. A equação de movimento da partícula é representado por (KENNEDY; EBERHART, 1995b):

$$\vec{x}(t+1) = \vec{x}(t) + \vec{v}(t+1) \quad (2.6)$$

onde $\vec{x}(t) \in R^n$ representa a posição de uma partícula p_i em um determinado tempo t (o tempo geralmente é uma iteração da execução) com uma adição de velocidade representada por $\vec{v}(t+1) \in R^n$. A velocidade no tempo $t+1$ possui uma equação que lhe representa e é definida como:

$$\vec{v}(t+1) = \omega \cdot \vec{v}(t) + C_1 \cdot \phi_1 \cdot (\vec{p}_{best}(t) - \vec{x}(t)) + C_2 \cdot \phi_2 \cdot (\vec{p}_{lider}(t) - \vec{x}(t)) \quad (2.7)$$

no qual, C_1 e C_2 são constantes que são usadas para delimitar a influência do elemento global e do elemento local no cálculo da velocidade da partícula. Já ϕ_1 e ϕ_2 , são números reais com valores entre 0 e 1 e que são determinados em cada iteração de maneira estocástica.

Além da abordagem feita pela equação 2.7, existe uma outra citada no trabalho de (CLERC; KENNEDY, 2002), na qual utiliza a fórmula de velocidade da partícula com uma pequena modificação. A modificação feita por (CLERC; KENNEDY, 2002), foi adicionar um fator de constrição que multiplica toda a equação 2.7, restringindo o crescimento ilimitado da velocidade das partículas. A equação modificada pode ser visualizada a seguir:

$$\vec{v}(t+1) = \chi \cdot (\omega \cdot \vec{v}(t) + C_1 \cdot \phi_1 \cdot (\vec{p}_{best}(t) - \vec{x}(t)) + C_2 \cdot \phi_2 \cdot (\vec{p}_{lider}(t) - \vec{x}(t))) \quad (2.8)$$

onde χ é representado pela fórmula:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|} \quad (2.9)$$

e φ por:

$$\varphi = C_1 + C_2 \quad (2.10)$$

O Algoritmo 1 mostra no geral como é o funcionamento do PSO. No começo, o enxame é inicializado. Nessa inicialização é incluída tanto as posições, quanto as velocidades. O representante *pbest* de cada partícula é inicializado e o líder é localizado. Logo em seguida são feitas as iterações até um número máximo, onde cada partícula irá voar através do espaço de busca e atualizar a sua posição, a do *pbest* e a posição do líder.

Segundo o trabalho de (REYES-SIERRA; COELLO, 2006), o MOPSO é uma estratégia baseada no PSO para a resolução de problemas de otimização multiobjetivo, no qual o esquema original do PSO foi alterado para suportar a resolução desse tipo de problema. No MOPSO, o algoritmo busca não somente uma solução (como é o caso do PSO), mas um conjunto de possíveis melhores soluções (Conjunto Pareto ótimo). No geral, para resolver problemas multiobjetivo, deve-se considerar as seguintes metas:

- i. Encontrar o maior número de elementos do conjunto Pareto ótimo;
- ii. Reduzir a distância entre a PF_{aprox} e a PF_{real} ;
- iii. Para ter uma melhor diversidade, tem que maximizar a disseminação das soluções encontradas, fazendo com que os vetores sejam distribuídos uniformemente.

Na geração da população é desejável encontrar diferentes soluções não-dominadas com apenas uma execução. Para os algoritmos evolutivos, aplicados em problemas de otimização multiobjetivo (por exemplo o MOPSO), deve ser levado em consideração três questões principais:

- i. Qual será o procedimento para selecionar as partículas líderes, lembrando de dar prioridade às soluções não-dominadas sobre aquelas dominadas?
- ii. Qual será o procedimento para manter as soluções não-dominadas durante todo o processo de busca, levando em consideração todas as populações passadas e não somente a população atual?
- iii. Qual será o procedimento para manter a diversidade do enxame, evitando a convergência para somente uma solução?

A fim de responder as perguntas, o algoritmo MOPSO pode dispor para cada partícula um conjunto de diferentes líderes, no qual apenas um irá ser escolhido para atualizar a sua

posição. Os líderes são mantidos em um lugar externo do enxame, no qual chamamos de arquivo externo (REYES-SIERRA; COELLO, 2006). Esse arquivo externo é um local de armazenamento de soluções não-dominadas que foram encontradas no momento da execução. As soluções não-dominadas que foram encontradas e que estão no arquivo serão utilizadas como líderes no momento que precisar fazer a atualização das posições das partículas do enxame. Além disso, o arquivo externo também geralmente serve como resultado de saída do algoritmo. Com o intuito de exibir como o algoritmo funciona, o Algoritmo 2 mostra as ações em comuns realizadas no PSO e no MOPSO, além de mostrar o diferencial que existe no MOPSO para a resolução dos MOPs. Logo a seguir é mostrado o Pseudocódigo do algoritmo MOPSO:

Algoritmo 2 Pseudocódigo do algoritmo MOPSO

```
1: Inicializa o enxame
2: Inicializa os líderes no arquivo externo
3: Qualidade(líderes)
4:  $g = 0$ 
5: while  $g < g_{max}$  do
6:   for Cada Partícula do
7:     Seleciona o líder
8:     Atualiza Posição(Vôo)
9:     Mutação
10:    Avaliação
11:    Atualiza o  $pbest$ 
12:  Atualiza os líderes no arquivo externo
13:  Qualidade(líderes)
14:   $g++$ 
15: Informa os resultados no arquivo externo
```

Depois de ter apresentado as equações de velocidade e de movimento, vamos a explicação do Algoritmo 2. Logo no início do algoritmo, ocorre a inicialização do enxame e dos líderes, lembrando que o conjunto desses líderes é um composto de soluções não-dominadas. Esse conjunto de líderes frequentemente é armazenado no arquivo externo e logo em seguida é feita uma análise da qualidade dos líderes com intenção de selecionar apenas um para cada partícula que compõe o enxame. Logo adiante é feita as iterações até um número máximo, onde cada partícula irá selecionar o líder e depois voar através do espaço de busca, atualizando a sua posição, avaliando a partícula e depois atualizando o $pbest$. No processo de iteração de cada partícula também é aplicado um operador de mutação, o qual é responsável por fornecer um mecanismo para o algoritmo fugir de ótimos locais e até acelerar a busca (PULIDO, 2005). Repetidamente, o algoritmo atualiza os líderes no arquivo externo e mede a qualidade deles. No final, as melhores soluções encontradas, ou seja, as soluções armazenadas no arquivo externo, são informadas.

Podem surgir alguns problemas quando tratamos da escolha e atualização dos líderes ou no controle do arquivo externo. Exemplo de problema causado na escolha de líderes é quando

tentamos selecionar apenas um líder de um conjunto composto de soluções não-dominadas, colocando ainda em pauta a questão se vamos escolher o líder aleatoriamente ou se vamos escolher algum critério que promova a diversidade. Já um exemplo de problema causado na gerência do arquivo externo, é que o arquivo pode ficar muito grande rapidamente e além disso ele tem que ser atualizado a cada geração, tornando a execução muito cara computacionalmente.

No trabalho de (MOSTAGHIM; TEICH, 2003), é proposto técnicas para resolução de problemas na escolha e atualização dos líderes. Nesse trabalho, os autores expõem técnicas criadas por outros autores, como meio para fazer comparações com a técnica proposta.

Dentre essas técnicas expostas, é citado a elaborada por (COELLO; LECHUGA, 2002) que tem como metodologia a construção de um mecanismo de forma que cada partícula possa selecionar um guia diferente. Esse mecanismo é baseado na criação de hipercubos que são fornecidos, dividindo o espaço de busca a ser explorado. Cada hipercubo recebe um valor *fitness*, dependendo da quantidade de partículas existentes no mesmo, sendo que, quando mais o número de partículas aumenta, mais é diminuído do valor *fitness*. Por último é escolhido o líder através do método de seleção chamado de roleta (COELLO; LAMONT; VELDHUIZEN, 2007), lembrando que esse método é baseado em seleção estocástica e que isso pode causar a seleção de um líder que não represente bem a escolha feita pela partícula.

Outra técnica é mostrada no trabalho de (FIELDSEND; SINGH, 2002). Essa técnica utiliza uma estrutura de dados chamada de *dominated tree* (FIELDSEND; SINGH, 2002), onde essa estrutura define uma ordem sobre membros de conjuntos de soluções não-dominadas, facilitando a escolha de um melhor indivíduo global para cada partícula do enxame, com a intenção de controlar a sua velocidade. Segundo (MOSTAGHIM; TEICH, 2003), a forma que os guias são selecionados em (FIELDSEND; SINGH, 2002) era melhor do que a forma exposta por (COELLO; LECHUGA, 2002). No entanto, ambos os trabalhos apenas testaram os algoritmos com somente duas funções objetivo, deixando a lacuna se os algoritmos seriam ainda eficientes quando o número de funções objetivo aumentasse.

Com a lacuna deixada nos trabalhos de (COELLO; LECHUGA, 2002) e (FIELDSEND; SINGH, 2002), o trabalho feito por (MOSTAGHIM; TEICH, 2003) apresentou um novo método chamado de *Sigma* para resolver o problema de encontrar o melhor guia local para cada partícula, testando o algoritmo com duas e três funções objetivo. Os resultados colhidos demonstraram uma boa convergência e diversidade. Na comparação com o método proposto por (FIELDSEND; SINGH, 2002), o *Sigma* saiu melhor na diversidade e na convergência. Já na comparação com o algoritmo *SPEA2* (ZITZLER et al., 2001) aplicados em duas funções objetivo, o *Sigma* saiu melhor na diversidade e na convergência, porém quando aplicados em problemas com três funções objetivo, o *Sigma* não obteve uma melhor convergência, mas obteve uma melhor diversidade.

É de grande interesse manter as características de preservação do elitismo e manter regras que são usadas para a delimitação do arquivo externo (LÓPEZ-IBÁÑEZ; KNOWLES;

LAUMANNNS, 2011). Por isso, o trabalho de (LÓPEZ-IBÁÑEZ; KNOWLES; LAUMANNNS, 2011) propõe técnicas para resolução de problemas no controle do arquivo externo. Nesse trabalho, o autor também expõem técnicas criadas por outros autores, como meio para fazer comparações.

De acordo com (LÓPEZ-IBÁÑEZ; KNOWLES; LAUMANNNS, 2011), uma forma simples de implementar um *Basic Archiver* (Arquivador Básico) é colocar o arquivo com tamanho fixo e preencher com soluções não-dominadas. Quando esse arquivo fica cheio é aplicada uma função de filtro que define qual solução irá ser removida para uma nova solução não dominada entrar. Logo, é dito que o arquivo externo tem tamanho máximo N e a função de filtro é utilizada quando o arquivo já está cheio, não havendo mais espaço para uma nova solução não-dominada, com isso, essa função tem o papel de selecionar qual das $N+1$ soluções vão permanecer no arquivo. Colocar o arquivo com tamanho fixo (ou seja, limitar o tamanho), faz com que a busca por líderes não se torne de alta complexidade, pois quanto maior o tamanho do arquivo, maior o custo durante a busca pelos líderes e isso pode influenciar negativamente a eficiência do algoritmo durante a busca.

O Arquivador do NSGA-II faz uma seleção em soluções não-dominadas, utilizando a *crowding distance*, fazendo assim uma estimativa do espaço vazio que rodeia a solução. O arquivador do NSGA-II também têm as características do *Basic Archiver*, mas a função de filtro utilizada no arquivador, basea-se na *crowding distance*, removendo a solução com o mínimo valor medido pela função (LÓPEZ-IBÁÑEZ; KNOWLES; LAUMANNNS, 2011). Outro arquivador relatado é o Arquivador Dominante (do inglês, *Dominating Archive*), no qual é um arquivador simples, onde somente adiciona, soluções que tenham dominado alguma outra solução do arquivo. O Arquivador Não Limitado (do inglês, *Unbound archiver*) também é citado, onde nesse método de arquivamento, não existe um limite para o arquivo externo.

O Arquivador Gride Adaptativo (do inglês, *Adaptive Grid*) distribui as soluções não-dominadas no espaço de objetivos, inserindo essas soluções em células distintas numa gride. O uso da gride é um instrumento que introduz diversidade nessas soluções, baseado na medição da densidade entre as regiões. O MGA (do inglês, *Multi-level Grid Archiving*) baseia-se no método de arquivamento anterior, mas são definidos blocos, que são representados por um índice. A ideia é que toda solução do arquivo deve pertencer a um bloco específico. Se duas ou mais soluções pertencerem a um mesmo bloco, somente é selecionado a solução mais próximo do índice do bloco. No geral, o artigo (LÓPEZ-IBÁÑEZ; KNOWLES; LAUMANNNS, 2011) demonstra exemplos de algoritmos que levam em consideração somente a convergência, outros visam principalmente a diversidade, alguns priorizam a qualidade geral de aproximação e por fim a proposta do artigo, fazendo a junção de características para conseguir ter uma melhor eficácia.

Uma técnica importante para levar em consideração é o controle da velocidade na qual as partículas dirigem-se para a fronteira de Pareto, pois a velocidade das partículas nestes algoritmos pode ser alta demais, implicando em movimentos errados. Com essa problemática

surgiu o SMPSO (do inglês, *Speed-constrained Multiobjective PSO*), tendo como principal característica limitar a velocidade das partículas (NEBRO et al., 2009). O algoritmo SMPSO possui características em comum com outros algoritmos multiobjetivo, como é o exemplo da utilização de um arquivo externo para o armazenamento de soluções não-dominadas, além de possuir um mecanismo de mutação para causar uma certa turbulência nas soluções com intenção de provocar uma maior diversidade e de utilizar a *Crowding Distance* para escolha do líder através do torneio binário (NEBRO et al., 2009).

2.2 Otimização com Muitos Objetivos

O número de melhores soluções cresce exponencialmente conforme o número de objetivos aumenta (JAIME; COELLO, 2009a). Com isso, a busca é dificultada porque os algoritmos enfrentam obstáculos na procura por melhores soluções, o que provoca um mal comportamento na seleção. Tudo isso causa a deterioração da busca e dificulta o progresso em direção às soluções ótimas. Esse comportamento é intensificado quando o número passa de 3 objetivos.

De acordo com (DEB; JAIN, 2014a) foram desenvolvidos algoritmos baseados na otimização evolutiva, mas que principalmente eram direcionados para problemas de otimização com geralmente dois ou três objetivos. Com a constante evolução na área da otimização multiobjetivo evolucionária (do inglês, *evolutionary multiobjective optimization* ou EMO (DEB; JAIN, 2014a)) houve a necessidade de melhoria nas técnicas para suportar as dificuldades causadas em problemas com muitos objetivos (ou seja, com quatro ou mais objetivos). Em problemas de otimização multiobjetivo com mais de três objetivos, é dito que são Problemas de Otimização com Muitos Objetivos (do inglês, *Many-Objective Optimization Problems* ou MaOPs).

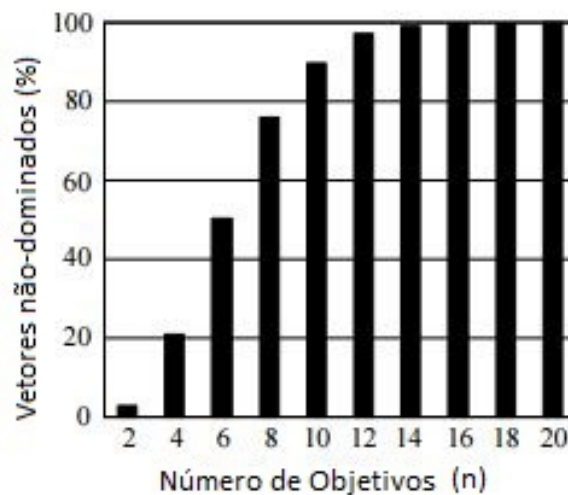
Para colocar em pauta as dificuldades enfrentadas pelos MOEAs, essa seção mostra alguns desafios enfrentados no mundo da Otimização com Muitos Objetivos. Segundo (DEB; JAIN, 2014a), as principais dificuldades enfrentadas são:

- i. **A maioria das soluções da população não são dominadas:** Com o crescente número de objetivos, torna-se cada vez maior o número de soluções não-dominadas. Como alguns MOEAs consideram a dominância de Pareto para a seleção, a busca pode ser um problema, já que a imposição de preferência é dificultada na seleção, não causando pressão a frente da fronteira de Pareto e deteriorando a convergência. Por isso, torna-se inviável a utilização desses algoritmos em problemas com muitos objetivos;
- ii. **Dimensão da fronteira de Pareto:** O aumento da dimensão do espaço de objetivos impacta diretamente no aumento exponencial de soluções não-dominadas e na necessidade de levar essas soluções a frente da fronteira de Pareto. Com isso, pode-se precisar de um número enorme de soluções para cobrir toda a frente da fronteira;

- iii. **Aumento da complexidade de processamento:** Quanto maior o número de objetivos, maior é o custo computacional para fazer as comparações com outras soluções, podendo aumentar ou diminuir exponencialmente o grau de computação, de acordo com o número de objetivos.

A fim de ilustrar a relação de porcentagem entre o número de objetivos e a quantidade de soluções não-dominadas, a imagem 3 demonstra esse exemplo de relação:

Figura 3 – Porcentagem de vetores não-dominados versus o número de objetivos.



Fonte: adaptada pelo autor

De acordo com (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008), para apresentar os resultados da relação entre a porcentagem de soluções não-dominadas e o número de objetivos, foram gerados de forma aleatória 200 vetores objetivos em um espaço com m -dimensões, onde $m = 2, 4, \dots, 20$ e m é o número de objetivos. Para cada m foram gerados 200 vetores objetivos e computado a porcentagem de vetores objetivos não-dominados existentes. Desse modo, é percebido o quanto o número de soluções não-dominadas aumenta quando o número de objetivos cresce.

2.2.1 Como resolver

Segundo (SCHÜTZE; LARA; COELLO, 2011), alguns desafios têm que ser dominados pelos algoritmos EMO em problemas de Otimização com Muitos Objetivos. Dentre esses desafios, está a capacidade de manter um bom agregado de soluções com objetivo de encaminhar a população para um local de interesse. Enfim, nesta seção é apresentado propostas para atenuar as dificuldades na computação de problemas de Otimização com Muitos Objetivos e que foram expostas na seção 2.2. Essas propostas estão disponíveis na literatura (ISHIBUCHI et al., 2011; CARVALHO, 2013; DEB; JAIN, 2014a) e podem ser classificados como:

- i. **Controle da Área de Dominância das Soluções:** em (SATO; AGUIRRE; TANAKA, 2007) foi utilizada uma outra abordagem de dominância. A metodologia adotada proporciona o comando do grau de expansão e compressão da área de dominância das soluções. Houve uma modificação da dominância de Pareto para melhorar a performance do NSGA-II. No geral o uso dessa relação de dominância trouxeram bons resultados, nos quais implicaram na redução de soluções não-dominadas e na pressão das soluções em frente a fronteira de Pareto;
- ii. **Redução do número de objetivos:** em (DEB; SAXENA, 2005; BROCKHOFF; ZITZLER, 2006) foram propostos métodos para reduzir o número de objetivos. O trabalho de (DEB; SAXENA, 2005) foi baseado na análise do componente, onde os objetivos desnecessários eram removidos, mas a forma da frente da fronteira de Pareto era mantida. Já o trabalho de (BROCKHOFF; ZITZLER, 2006) foi baseado na dominância de Pareto, em que os objetivos eram removidos quando não houvesse alteração na relação de dominância entre as soluções;
- iii. **Uso de decomposição:** é dito que método de decomposição é a técnica usada para decompor um problema multiobjetivo em sub-problemas. Esses subproblemas são problemas escalares que podem ser otimizados por MOEAs (ZHANG; LI, 2007);
- iv. **Uso de preferências:** o número de soluções aumenta exponencialmente de acordo com o aumento do número de objetivos. Com isso, pode ser interessante manter o foco em um determinado local da fronteira de Pareto, e para isso é usado os indicadores de preferência (DEB; SUNDAR, 2006; NEBRO et al., 2009);
- v. **Outros métodos de avaliação de *fitness*:** outras técnicas de avaliação de *fitness* são propostas para medir o desempenho do conjunto de soluções. Deixando de utilizar o sistema de avaliação do *fitness* baseado na dominância de Pareto e passando a usar outros algoritmos como exemplo o hipervolume, epsilon e outros mais (ZITZLER; THIELE, 1998; ZITZLER; KÜNZLI, 2004).

Com a apresentação das propostas para atenuação das dificuldades encontradas nos MaOPs, será mostrado na próxima seção alguns trabalhos correlacionados com esse tema.

2.2.2 Trabalhos relacionados aos MaOPs

Vários trabalhos na literatura tentam resolver problemas da categoria dos MaOPs. Nessa seção, são expostos alguns desses trabalhos que adotam estratégias para a resolução destes tipos de problemas. Dentre esses trabalhos, será mostrado os que se destacam no uso da técnica de Controle da Área de Dominância das Soluções (SATO; AGUIRRE; TANAKA, 2007), no uso de arquivamento para controlar a convergência e diversidade (BRITTO; POZO, 2012b), baseado em múltiplos enxames (CAMPOS; POZO; DUARTE, 2013), no uso de múltiplos enxames +

arquivadores (BRITTO; MOSTAGHIM; POZO, 2014), na utilização de pontos de referência (BRITTO; POZO, 2014) além do trabalho baseado em múltiplos enxames + pontos de referência (BRITTO; POZO, 2015).

Em (SATO; AGUIRRE; TANAKA, 2007) é apresentado um método que visa controlar a área de dominância de soluções. A metodologia adotada proporciona o comando do grau de expansão e compressão da área de dominância das soluções através de parâmetros passados pelo usuário e na alteração do *fitness* de cada solução para movimentar a solução pelo espaço de objetivos. Quando se controla a área de dominância, a relação de dominância é modificada e isso implica numa relação diferente da dominância convencional. Com isso, a distribuição das soluções a frente da fronteira é melhorada. De acordo com a manipulação do parâmetro responsável por expandir ou fazer a compressão da área, o número de soluções não-dominadas pode aumentar ou diminuir. Isso acaba sendo importante para a convergência em MaOPs, já que o número de soluções não-dominadas cresce com o número de objetivos, provocando a deterioração do algoritmo, uma vez que a maioria das soluções são não-dominadas, dificultando assim a procura por melhores soluções. Foi realizado o estudo em cima de problemas para verificar a performance da busca. O *benchmark*³ de estudo utilizado foi o *multiobjective 0-1 knapsack problems*⁴. Em geral, o algoritmo proposto teve melhor diversidade e convergência do que o uso da dominância convencional e como trabalho futuro foi proposto uma análise para tornar o parâmetro de controle da área de dominância adaptável, principalmente em problemas que possuam características pouco conhecidas.

O trabalho de (BRITTO; POZO, 2012b) utilizou estratégias de arquivamento para controlar a diversidade e convergência em MaOPs. O efeito negativo da deterioração da busca foi diminuído através dos métodos de arquivamento usados no algoritmo MOPSO e que foram apresentados no trabalho. Neste, foram usados métodos de arquivamento disponíveis na literatura e também foram propostos novos métodos, nos quais são chamados de: *Ideal Archiver*, *Distributed Archiver* and *Distance to Reference Points Archiver*. O algoritmo MOPSO foi aplicado ao problema DTLZ2 (DEB et al., 2002) e utilizou indicadores de qualidade (JAIMES; COELLO, 2009b) para investigar como os métodos de arquivamento afetavam a diversidade e a convergência. Na avaliação dos resultados foi percebido que alguns métodos se destacaram. Em geral, a não limitação do tamanho do arquivo gera bons resultados, porém isso é impraticável em problemas reais. Dos métodos de arquivamento utilizados, o arquivador Ideal obteve bons resultados em termos de convergência e o arquivador MGA em termos de diversidade. Além disso, os arquivadores baseados na dominância epsilon também se destacaram. Os trabalhos futuros incluíam fazer comparações entre o MOPSO com outros algoritmos na temática dos MaOPs e testar com diferentes *benchmarks*.

Em (CAMPOS; POZO; DUARTE, 2013), foi aplicado a abordagem de múltiplos enxames

³ é um conjunto de operações responsável por avaliar o desempenho

⁴ são problemas básicos de otimização combinatória (ZITZLER; THIELE, 1998)

(do inglês, *Multi-swarm*) com uma estratégia de comunicação assíncrona entre os mesmos para resolver Problemas de Otimização com Muitos Objetivos. A técnica de *Multi-swarm* (CAMPOS; POZO; DUARTE, 2013) baseia-se em subpopulações de partículas independentes que trocavam experiências entre elas, onde cada subpopulação era direcionada de forma individual pelo seu guia, ou seja, pela melhor partícula da subpopulação e também pela melhor partícula encontrada em todos os enxames. Lembrando que o direcionamento feito pelos guias tem como o objetivo encontrar as melhores soluções, nas quais são as próximas à fronteira de Pareto. A forma de comunicação utilizou um tipo de arquivo, onde a cada iteração era preenchido de soluções não-dominadas, sendo que cada enxame possuía individualmente uma estrutura como essa. Em cada iteração o arquivo era enviado para os outros enxames vizinhos e logo depois era esvaziado. A estratégia de comunicação assíncrona também utiliza o arquivo para enviar as melhores soluções para cada enxame. O envio das soluções contidas no arquivo era através da propagação delas para cada enxame, usando uma topologia onde mantinham todos conectados. Quando o arquivo é propagado para outro enxame, o enxame que recebeu terá as soluções não-dominadas do arquivo local mais o recebido do outro enxame. Com isso uma atualização é necessária para não encher o arquivo local, então é feito o procedimento de verificação de quais soluções são dominadas entre os dois arquivos. O algoritmo proposto foi testado em dois problemas, o DTLZ2 e DTLZ4 (DEB et al., 2002), já que são problemas com muitos objetivos em que a fronteira de Pareto é conhecida. Através de uma análise empírica dos resultados, baseados em métricas convencionais, ficou percebido que no geral a busca no espaço de soluções com múltiplos enxames trazia um efeito positivo na convergência e diversidade.

No trabalho de (BRITTO; MOSTAGHIM; POZO, 2014) também foi utilizado a abordagem de *Multi-swarm* para resolver problemas com mais de três objetivos. Foi proposto um algoritmo que obtenha boa convergência e boa diversidade. O trabalho utilizou dois métodos de arquivamento para o armazenamento das soluções não-dominadas dos enxames, o arquivador chamado de *Ideal* e o outro de *Multi-level Grid Archiving* (MGA). O *Ideal* incrementa a convergência das soluções não-dominadas para a PF_{real} , enquanto que o MGA obtém boa diversidade de soluções (BRITTO; POZO, 2012b). A ideia por trás do método proposto é combinar diferentes métodos de arquivamento através de uma estratégia de múltiplos enxames. Os vários enxames são conectados numa topologia de anel e semelhante ao trabalho (CAMPOS; POZO; DUARTE, 2013) passam as soluções de um arquivo para outro. A diferença é que um enxame que usa o arquivador *Ideal* passa suas soluções para um enxame com arquivador MGA. Foram criados dois algoritmos, o **A-Multi Conv** e o **A-Multi Div**. No qual o **A-Multi Conv** obteve bons resultados em termos de convergência e o **A-Multi Div** obteve bons resultados em termos de diversidade. O que geralmente diferenciava o resultado entre os dois algoritmos era o ajuste de parâmetros de configuração do número de *swarms*, partículas e atualização delas em cada iteração. Com a mudança apenas nos parâmetros de configuração, os algoritmos passaram a se chamar somente **A-Multi**. O **A-Multi** foi testado nos problemas da família DTLZ (DEB et al., 2002) (os problemas testados foram o DTLZ2, DTLZ4 e DTLZ6) e comparados com

outros algoritmos como o NSGA-II, SMPSO e o I-MOPSO (BRITTO; POZO, 2014; BRITTO; POZO, 2012a). Houve algumas limitações como o uso fixo dos parâmetros de configuração e o não uso de outras topologias de comunicação (o trabalho somente usou a topologia de anel (REYES-SIERRA; COELLO, 2006)), mas no geral, através dos resultados notáveis em termos de convergência e diversidade, ficou percebido que o **A-Multi** é melhor do que os algoritmos comparados e que utilizar múltiplos enxames é um bom caminho a se seguir.

Em (BRITTO; POZO, 2014) foi aplicado pontos de referência com o intuito de atualizar o arquivo em algoritmos MOPSO aplicados à Problemas com Muitos Objetivos. O atual trabalho teve como objetivo estender o apresentado em (BRITTO; POZO, 2012a), onde foi proposto o algoritmo chamado **I-MOPSO**. O **I-MOPSO** tinha como característica introduzir mais convergência para PF_{real} através de um método de arquivamento e incorporar mais diversidade nas soluções por meio de um método de seleção de líderes. O algoritmo **I-MOPSO** mostrou uma boa convergência, diversidade e também denotou a aproximação das soluções aos pontos de referência. Com isso, foi proposto uma extensão do **I-MOPSO** em (BRITTO; POZO, 2014) e o algoritmo passou a se chamar **REF-I-MOPSO**. Esse método de arquivamento se baseava em pontos de referência que guiavam as soluções para o espaço onde possivelmente teriam pontos ideais (BRITTO; POZO, 2014). Um diferencial dessa extensão foi a incorporação de outro método de arquivamento, no qual se chama *Hyperplane Archiver*, onde ele guardava no arquivo as soluções que estavam mais próximas dos pontos de referência e quando o arquivo fosse totalmente preenchido, a solução que estivesse mais distante era retirada do arquivo. O **REF-I-MOPSO** também foi testado nos problemas da família DTLZ (DEB et al., 2002) (foram o DTLZ2, DTLZ4 e DTLZ6) e comparado com outros algoritmos como o I-MOPSO, SMPSO e o I-Sigma (MOSTAGHIM; TEICH, 2003). Com a análise dos indicadores percebeu-se que em termos de convergência o **I-Sigma** foi melhor que os outros, já na diversidade o **REF-I-MOPSO** se destacou sobre os outros, mas foi bastante semelhante ao **I-MOPSO**. Como trabalhos futuros, os autores sugerem uma abordagem utilizando múltiplos enxames.

O trabalho de (BRITTO; POZO, 2015) propôs um novo algoritmo baseado em múltiplos enxames, no qual foi chamado de **R-Multi**. O **R-Multi** foi implementado para resolver algumas limitações que os MOEAs possuem, como é o caso da dificuldade de encontrar bons resultados quando o número de funções objetivo é maior que três. Nessa temática, o trabalho citou várias abordagens que são importantes para a melhoria dos resultados, como por exemplo a adição de relações de preferência, no qual tem função de guiar as partículas para uma possível melhor região da fronteira de Pareto. O **R-Multi** utilizou a abordagem de múltiplos enxames, onde cada enxame usava um *Hyperplane Archiver* e executava o MOPSO. Um diferencial do **R-Multi** é que cada enxame utiliza de pontos de referência fixos, no intuito de provocar a convergência das partículas para uma região específica da fronteira de Pareto. O **R-Multi** também foi testado nos problemas da família DTLZ (DEB et al., 2002) (o DTLZ2, DTLZ4 e DTLZ6) e comparados com outros algoritmos como o **I-Multi** (BRITTO; MOSTAGHIM; POZO, 2013), **A-Multi** (BRITTO; MOSTAGHIM; POZO, 2014) e o MOEA/D-DRA (ZHANG; LIU; LI, 2009). Para a avaliação

dos resultados foi feito uma análise empírica, onde se constatou que o **R-Multi** obteve uma boa diversidade e controlou a deterioração da busca em um número alto de funções objetivo. Como trabalhos futuros, é sugerido a exploração da abordagem de diferentes pontos de referência e a combinação de outros métodos de arquivamento com o *Hyperplane Archiver*.

O trabalho de (CHENG et al., 2016) é apresentado o algoritmo RVEA (Reference Vector Guided Evolutionary Algorithm), onde é comparado com o NSGA-III de Deb e Jain (2014a) e outros algoritmos evolutivos, mostrando-se competitivo quando aplicados à problemas de otimização com muitos objetivos. O RVEA possui um conceito parecido ao NSGA-III, mas substitui a seleção baseada em dominância de Pareto do NSGA-III por uma função de distância angular-penalizada, a qual tem como característica equilibrar a diversidade e convergência em problemas com alta dimensionalidade. Além disso, enquanto a NSGA-III renormaliza os objetivos em cada iteração, o RVEA escala periodicamente os vetores de referência, reduzindo de maneira considerável a sobrecarga do algoritmo. O vetor de referência pode decompor um problema com muitos objetivos em sub-problemas, além de indicar e segmentar um subconjunto para uma região específica da fronteira de Pareto.

O trabalho de (LI et al., 2015) foi aplicado estratégias de dominância e decomposição com finalidade de manter um equilíbrio entre os fatores de convergência e diversidade aplicado em MaOPs. Com a junção dessas duas abordagens, surgiu o chamado MOEA/DD (Multiobjective Evolutionary Algorithm Based on Dominance and Decomposition). O algoritmo citado gera vetores de peso no qual são largamente disseminados num espaço de objetivos com alta dimensão, onde cada vetor de peso define um sub-problema e também especifica uma sub-região única no espaço de objetivos.

2.3 Conclusões

Foi discutido o funcionamento geral dos algoritmos e os objetivos que cada um deles possuem, no qual geralmente é citado a convergência e a diversidade de soluções à frente da fronteira de Pareto. A ideia do algoritmo foi rapidamente explicada e mostrado os resultados encontrados, em quais problemas de *benchmarking* foram utilizados, além de mostrar com quais algoritmos foram comparados para a análise dos resultados. Dessa forma, depois que os resultados foram analisados, ficaram em destaque algumas limitações.

Em (SATO; AGUIRRE; TANAKA, 2007) o método proporciona o comando do grau de expansão e compressão da área de dominância das soluções através de parâmetros passados pelo usuário e na alteração do *fitness* de cada solução para movimentar a mesma pelo espaço de objetivos. Com isso, uma limitação desse método está no uso de parâmetros constantes, ou seja, talvez o uso de parâmetros dinâmicos tragam melhores resultados. Além disso, pode ser feita uma combinação entre a abordagem feita pelo algoritmo e a incorporação de preferências. O trabalho de (BRITTO; POZO, 2012b) utilizou estratégias de arquivamento para controlar a

diversidade e convergência em MaOPs, mas é fato que alguns métodos tiveram bons resultados em termos de convergência e outros em diversidade. Logo, não foi proposto um método único que apresente um bom resultado em ambos os aspectos. Em (CAMPOS; POZO; DUARTE, 2013), foi aplicado a abordagem de múltiplos enxames com uma estratégia de comunicação assíncrona entre os mesmos para resolver Problemas de Otimização com Muitos Objetivos, porém a limitação desse é que ele focou na topologia mas não trabalhou diferentes arquivadores. No trabalho de (BRITTO; MOSTAGHIM; POZO, 2014) também foi utilizado a abordagem de múltiplos exames para resolver problemas com mais de três objetivos, os já conhecidos MaOPs, entretanto, percebeu-se algumas limitações, como o uso fixo dos parâmetros de configuração e o não uso de outras topologias de comunicação. Em (BRITTO; POZO, 2014) foi aplicado pontos de referência com o intuito de atualizar o arquivo em algoritmos MOPSO aplicados à Problemas com Muitos Objetivos, porém, o trabalho poderia melhor explorar a utilização do *Hyperplane Archiver* em problemas com diferentes formas da fronteira de Pareto, podendo também utilizar a abordagem de múltiplos enxames para possivelmente obter melhores resultados. O trabalho de (BRITTO; POZO, 2015) utilizou a abordagem de múltiplos enxames, onde cada enxame usava um *Hyperplane Archiver* e executava o MOPSO. Entretanto, não trabalha de forma adequada com pontos de referência, no qual os pontos utilizados eram fixos. Com a exposição dessas limitações, a Seção a seguir mostra a incorporação de algumas características que se destacaram para sugerir novos esquemas.

Com isso, o trabalho tem com o objetivo explorar as limitações expostas pelos algoritmos relacionados aos MaOPs, propondo assim novos esquemas. As limitações devem ser consideradas e trabalhadas para levar a uma melhoria do funcionamento do algoritmo no uso de problemas com muitos objetivos. Essas limitações foram extraídas de trabalhos relacionados que utilizaram diferentes abordagens para resolução dos MaOPs. Desse forma, são propostos novos algoritmos baseados em múltiplos enxames aplicados nesses tipos de problemas no capítulo 3.

3

Algoritmos de Múltiplos Enxames Aplicados a MaOPs

Múltiplos Enxames ([CAMPOS; POZO; DUARTE, 2013](#)) baseia-se em subpopulações de partículas independentes que trocam experiências entre elas, onde cada subpopulação é direcionada de forma individual pelos guias, ou seja, pela melhor partícula da subpopulação e a melhor de todos os enxames. Lembrando que o direcionamento feito pelos guias tem como o objetivo encontrar as melhores soluções, nas quais são as próximas a fronteira de Pareto. Segundo [Campos, Pozo e Duarte \(2013\)](#), [Britto, Mostaghim e Pozo \(2014\)](#), [Britto e Pozo \(2015\)](#) e contextualizando com o Capítulo 2, a abordagem de *Multi-swarm* é um bom caminho a ser seguido e os algoritmos baseados no MOPSO, métodos de arquivamento e pontos de referência se destacam devido aos melhores resultados encontrados quando comparados à outras meta-heurísticas.

Nesse capítulo será demonstrado a construção de algoritmos que incorporam as características de múltiplos enxames com topologias, arquivamento e pontos de referência. Na seção 3.1 é mostrado a construção e funcionamento do algoritmo MSAT, destacando sua característica marcante da combinação de arquivadores associados a topologias. Na seção 3.2 é apresentado o algoritmo MSAT-H, onde é uma extensão do MSAT utilizando o hiperplano e pontos de referência que são gerados dinamicamente. Já na seção 3.3 é apresentado o algoritmo MSAT-MH, onde é uma extensão do MSAT-H que além de utilizar o hiperplano, pontos de referência que são gerados dinamicamente, possui funções matemáticas que modificam o hiperplano com intenção de adaptar/moldar o algoritmo para determinado tipo de problema.

3.1 Algoritmo MSAT

O Algoritmo *Multi-Swarm* baseado em Arquivamento e Topologias (MSAT) tem como característica principal o uso do MOPSO com múltiplos enxames para resolver MaOPs. Ele combina a estratégia de usar métodos de arquivamento e diferentes topologias de comunicação visando obter uma boa convergência e diversidade quando aplicada a problemas com muitos objetivos.

O MSAT é composto por diferentes enxames, cada um com sua própria população e arquivo externo. Cada arquivo externo tem seu próprio método de arquivamento. Durante a busca, em um dado momento os enxames trocam informações. A comunicação entre os diferentes enxames é definida por uma topologia predefinida. As informações trocadas são as soluções apresentadas em cada arquivo de soluções não denominadas. Quando um enxame recebe um conjunto de soluções não denominadas, ele não apaga seu próprio arquivo, em vez disso, ele tenta adicionar as novas soluções ao seu arquivo usando o método definido pelo enxame.

Algoritmo 3 mostra as principais etapas executadas pelo MSAT. O MSAT tem quatro parâmetros de entrada principais: os métodos de arquivamento selecionados, o número de enxames, a topologia de comunicação e o momento da comunicação. Estes parâmetros serão explicados mais adiante nesta seção.

Primeiro, os parâmetros de entrada são definidos. O número de enxames indica quantos enxames serão executados e se comunicarão. Cada enxame executa um algoritmo MOPSO independente. Assim, além dos parâmetros principais, todo o parâmetro MOPSO básico deve ser definido, como tamanho da população, tamanho do arquivo e coeficientes de pesquisa. SMPSO (NEBRO et al., 2009) é usado como base para cada algoritmo MOPSO. A topologia é escolhida como padrão de comunicação, definindo como cada enxame trocará seu líder. O momento da comunicação indica quando, durante a busca, a comunicação ocorrerá. A combinação de arquivos indica o método de arquivamento utilizado por cada enxame.

Algoritmo 3 Algoritmo MSAT

```

1: Entrada:
2:   Número de enxames e topologia de comunicação;
3:   Combinação de arquivadores;
4:   Momento de comunicação;
5:   _____
6: Inicializa os enxames;
7: Avaliação(todas as partículas nos enxames);
8: Inicializa os líderes nos arquivos externos;
9:  $g = 0$ ;
10: while A condição final não for alcançada do
11:   for Cada enxame do
12:     for Cada partícula no enxame do
13:       Selecione o líder;
14:       Atualiza a posição;
15:       Mutação;
16:       Avaliação;
17:       Atualiza pbest;
18:     Atualizar os líderes do enxame no arquivo externo;
19:   if Momento de comunicação entre enxames then
20:     if Topologia de anel then
21:       Faz comunicação com a topologia de anel;
22:     else if Topologia de grafo completo then
23:       Comunicação com a topologia de grafo completo;
24: Saída:
25:   Conjunto de soluções não-dominadas entre todos os enxames;

```

No início do algoritmo, ocorre a inicialização dos enxames. Nesta etapa, cada enxame usa o procedimento de inicialização básico executado pelo algoritmo MOPSO. Após o procedimento de inicialização, cada partícula é avaliada e os líderes dos enxames são selecionados de acordo com os métodos de arquivamento. O próximo passo é o *loop* de pesquisa. Até que uma condição final seja cumprida (aqui é usado um número máximo de iteração), cada enxame irá executar o movimento de suas partículas: cada partícula irá selecionar o líder; irá voar através do espaço de busca, atualizando sua posição; irá avaliar sua nova posição; e finalmente irá atualizar a sua melhor partícula (*pbest*). No processo de iteração de cada partícula também é aplicado um operador de mutação (turbulência). Esta mutação é responsável por fornecer um mecanismo para o algoritmo para escapar do ótimo local e até mesmo acelerar a pesquisa. No final do procedimento de posição de atualização de enxame, os líderes do enxame são atualizados através do método de arquivamento. Depois que cada enxame se moveu, o algoritmo deve verificar se é hora de executar a comunicação. Se for verdade, os enxames devem realizar a troca das soluções arquivadas, de acordo com a topologia de comunicação. O MSAT fornece duas topologias diferentes: anel ou grafo completo, conforme é mostrado na Figura 4.

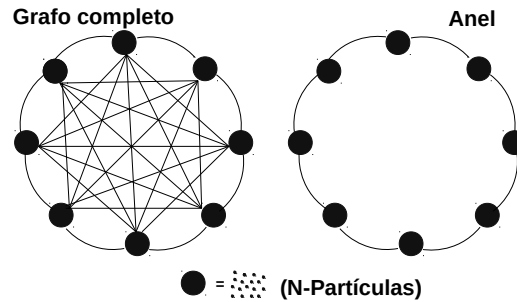


Figura 4 – Múltiplos enxames com diferentes topologias.

Após a comunicação, a condição final é verificada e o *loop* de pesquisa pode ser iniciado novamente. Se a condição de fim for atendida, as melhores soluções encontradas pelo MSAT é o conjunto de todas as soluções não-dominadas entre todos os arquivos dos enxames.

3.2 Hiperplano

O algoritmo MSAT foi modificado e acrescentado característica de pontos de referência, além da característica de topologia e arquivamento com múltiplos enxames que foram implementados no MSAT. O novo algoritmo se chama MSAT-H, pois possui características semelhantes ao MSAT, com exceção da particularidade de se basear e gerar dinamicamente um hiperplano com diferentes pontos de referência que guiarão as partículas para possivelmente um melhor local em frente à fronteira de Pareto.

O hiperplano estabelece um conjunto de pontos que cobrem uma área maior do espaço de objetivos. Ele pode ser elaborado de forma automática, apenas utilizando os pontos que definem os extremos, além do espaço entre cada ponto de referência. O hiperplano pode ser gerado utilizando a ideia proposta por [Deb e Jain \(2014b\)](#) com o NSGA-III. Nesse algoritmo, é usado um conjunto de pontos de referência que são inseridos num hiperplano normalizado que é igualmente inclinado a todos os eixos objetivos e tem uma intersecção em 1 para cada eixo. Por exemplo, um hiperplano com 3 objetivos intercepta (1,0,0), (0,1,0) e (0,0,1), conforme mostra a Figura 5.

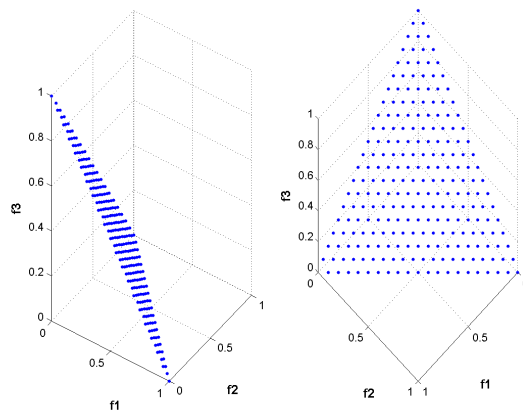


Figura 5 – Imagem de [Oliveira e Carvalho \(2017\)](#) que mostra um exemplo de hiperplano normalizado aplicado em 3 objetivos.

Com esse mecanismo, pode-se melhorar a diversidade do algoritmo de acordo com a escolha dos pontos de referência. Como exemplo da escolha aleatória dos pontos de referência entre todos os pontos do hiperplano. Com isso, o algoritmo favorece a diversidade ao invés de convergência.

No geral, a particularidade a ser destacada para o algoritmo MSAT-H é a geração do hiperplano, a quantidade e uso de pontos de referência, onde esses são gerados dinamicamente no sentido de carregar pontos diferentes e permitir uma maior exploração do espaço de busca.

Depois de exemplificar o hiperplano com seus pontos de referência, segue agora o pseudocódigo do MSAT-H:

Algoritmo 4 Algoritmo MSAT-H

```

1: Entrada:
2:   Número de enxames e topologia de comunicação;
3:   Combinação de arquivadores;
4:   Momento de comunicação;
5:   Quantidade de pontos de referência;
6:   _____
7: Inicializa os enxames;
8: Avaliação(todas as partículas nos enxames);
9: Inicializa os líderes nos arquivos externos;
10: Inicializa os pontos de referência;
11:  $g = 0$ ;
12: while A condição final não for alcançada do
13:   for Cada enxame do
14:     for Cada partícula no enxame do
15:       Selecione o líder e um ponto de referência;
16:       Atualiza a posição;
17:       Mutação;
18:       Avaliação;
19:       Atualiza  $pbest$ ;
20:     Atualizar os líderes do enxame no arquivo externo;
21:   if Momento de comunicação entre enxames then
22:     if Topologia de anel then
23:       Faz comunicação com a topologia de anel;
24:     else if Topologia de grafo completo then
25:       Comunicação com a topologia de grafo completo;
26: Saída:
27:   Conjunto de soluções não-dominadas entre todos os enxames;

```

Como pode ser visto no Algoritmo 4, a implementação é muito parecida com a do MSAT (características diferentes do novo algoritmo em relação ao anterior são mostradas na cor azul), tendo vários parâmetros em comum. Lembrando que no MSAT-H cada partícula do enxame irá ser guiada por um líder até um ponto de referência, no qual esse ponto será específico para todas as partículas do enxame.

3.3 Modificações no Hiperplano

De acordo com o trabalho de Oliveira e Carvalho (2017) a principal contribuição do NSGA-III para resolver o MaOPs está em seus pontos de referência. No entanto, os pontos de referência ainda não foram totalmente investigados. Apesar do problema, o hiperplano é sempre o mesmo, variando apenas em termos do tamanho do conjunto de pontos. Além disso, embora o RVEA (CHENG et al., 2016) seja um algoritmo que lide com transformações de pontos de referência, ele apenas ajusta dinamicamente seus pontos de distribuição para melhor representar

a solução. Contudo, nunca se altera a forma do hiperplano.

O algoritmo MSAT-MH foi criado para investigar se modificações no hiperplano poderia trazer um melhor comportamento e melhores resultados do que as combinações anteriores. Foram adotadas as mesmas regras do algoritmo MSAT-H, com exceção da característica de modificação do hiperplano. Foi utilizado funções matemáticas senoidais, côncavas e convexas baseada em [Oliveira e Carvalho \(2017\)](#) no intuito de mudar a forma do hiperplano e identificar quais funções se comportam melhor para cada tipo de problema.

Na transformação Côncava, a ideia básica dessa transformação é "esticar" o hiperplano original em um formato côncavo. A Figura 6 exibe a saída após a transformação. Observe que ela tem um formato esférico. Essa distribuição desigual pode ser facilmente corrigida com o uso da adaptação guiada de vetores proposto por [Cheng et al. \(2016\)](#), já que ela pode mover os pontos para melhor representar a população.

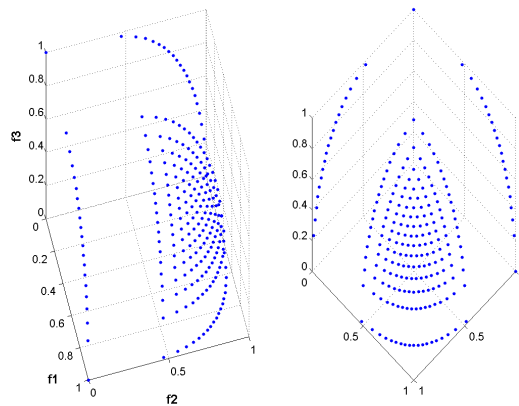


Figura 6 – Imagem de [Oliveira e Carvalho \(2017\)](#) que mostra o hiperplano em um formato côncavo.

A transformação Convexa é similar à transformação anterior, a ideia básica é "esticar" o hiperplano em um formato convexo. Essa transformação foi baseada na transformação do problema DTLZ2 para o DTLZ2 Convexo (ou CDTLZ2) apresentado em [\(DEB; JAIN, 2014a\)](#). A Figura 7 mostra uma representação 3D do hiperplano transformado. Observe que ele se aproxima rapidamente da origem, e que tem uma grande densidade de pontos em suas proximidades.

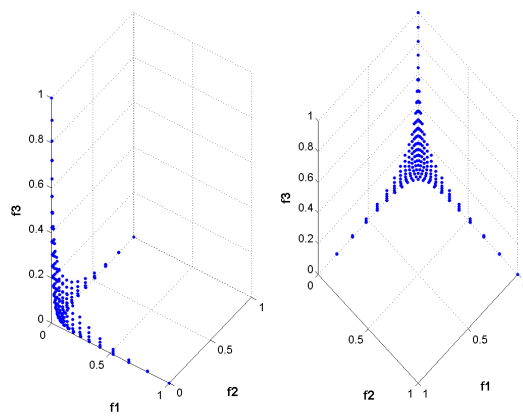


Figura 7 – Imagem de [Oliveira e Carvalho \(2017\)](#) que mostra o hiperplano em um formato convexo.

Já a transformação Senoidal, estica o hiperplano e deixa em um formato senoidal. A Figura 8 mostra a saída do algoritmo. Observe que ela tem 3 partes côncavas, e 1 convexa no centro próxima a origem.

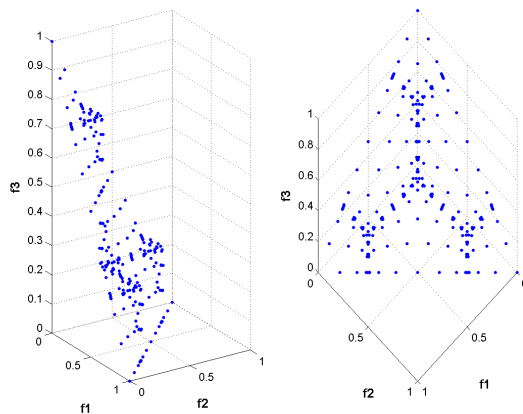


Figura 8 – Imagem de [Oliveira e Carvalho \(2017\)](#) que mostra o hiperplano em um formato senoidal.

Depois de exemplificar o comportamento e forma do hiperplano quando são aplicadas as funções matemáticas, segue agora o pseudocódigo do MSAT-MH:

Algoritmo 5 Algoritmo MSAT-MH

```

1: Entrada:
2:   Número de enxames e topologia de comunicação;
3:   Combinação de arquivadores;
4:   Momento de comunicação;
5:   Quantidade de pontos de referência;
6:   Tipo da função de transformação;
7:   _____
8:   Inicializa os enxames;
9:   Avaliação(todas as partículas nos enxames);
10:  Inicializa os líderes nos arquivos externos;
11:  Inicializa os pontos de referência;
12:  Função de transformação aplicada nos pontos de referência;
13:   $g = 0$ ;
14:  while A condição final não for alcançada do
15:    for Cada enxame do
16:      for Cada partícula no enxame do
17:        Selecione o líder e um ponto de referência;
18:        Atualiza a posição;
19:        Mutação;
20:        Avaliação;
21:        Atualiza  $pbest$ ;
22:      Atualizar os líderes do enxame no arquivo externo;
23:    if Momento de comunicação entre enxames then
24:      if Topologia de anel then
25:        Faz comunicação com a topologia de anel;
26:      else if Topologia de grafo completo then
27:        Comunicação com a topologia de grafo completo;
28:  Saída:
29:  Conjunto de soluções não-dominadas entre todos os enxames;

```

De acordo como pode ser visto no Algoritmo 5, a implementação é muito parecida com a do MSAT-H (características diferentes do novo algoritmo em relação ao anterior são mostradas na cor azul), possuindo vários parâmetros em comum. A particularidade a ser destacada é o tipo de função de transformação que será aplicada nos pontos de referência, onde provavelmente dependendo do tipo de função e do tipo de problema o algoritmo possa comportar-se de forma esperada e atingir convergência e diversidade em frente à fronteira.

4

Experimentos

Este capítulo descreve os procedimentos de avaliação dos algoritmos propostos nesse trabalho. Como o algoritmo proposto têm a característica de combinar métodos de arquivamento, topologias e pontos de referência, os principais objetivos do conjunto de experimentos serão observar quais métodos, quantidade de pontos e topologias geram melhores resultados em cenários de muitos objetivos. Além dessa análise, as melhores configurações de MSAT são confrontadas com o algoritmo NSGA-III ([DEB; JAIN, 2014a](#)). As próximas seções apresentarão a metodologia experimental, os problemas, indicadores de qualidade, os experimentos e discutirão seus resultados.

4.1 Metodologia

Serão três conjuntos de experimentos, cada um validando um algoritmo proposto. A ideia é mostrar passo-a-passo e de forma objetiva o que será avaliado em cada experimento. O primeiro algoritmo desenvolvido foi o MSAT, no qual serviu de base para a construção do outros algoritmos, onde nesse experimento foi levado como principal avaliação quais seriam as melhores combinações de arquivadores. O segundo algoritmo implementado foi o MSAT-MH, ele adotava as funções do MSAT com a distinção de seguir como principal avaliação o uso pontos de referência e variar esses pontos para possivelmente descobrir alguma peculiaridade que podia ser interessante. O terceiro e último experimento foi com o algoritmo MSAT-MH, ele possui os atributos do MSAT-H, além de ter como análise principal a modificação dos pontos de referência através de funções matemáticas, onde de certa forma essas funções têm como característica adaptar/moldar esses pontos para cada tipo de fronteira de Pareto.

4.2 Problemas

Os problemas de *benchmarking* utilizados são da família DLTZ (DEB et al., 2002). São utilizados problemas onde a verdadeira frente Pareto-ótima é conhecida analiticamente. Para cada problema, a variável k representa a complexidade da pesquisa, onde $k = n - m + 1$ (n número de variáveis, m número de objetivos). Neste estudo, os algoritmos são aplicados a problemas com 3, 5, 10 funções objetivo e $k = 10$. O problema DTLZ2 pode ser usado para investigar a habilidade dos algoritmos para escalar seus desempenhos em grande número de objetivos. Este problema tem forma côncava da fronteira de Pareto. O problema DTLZ4 é uma extensão de DTLZ2 e é usado para investigar a capacidade de manter uma boa distribuição de soluções. O problema DTLZ4 gera mais soluções perto do plano $f_m - f_1$, então os algoritmos focalizam suas soluções nessa região. DTLZ6 é uma variação do DTLZ2, onde a fronteira Pareto-ótimo é definida por uma curva, em vez de uma esfera. Além disso, este problema apresenta $(3^k - 1)$ Pareto-ótimo local. O DTLZ2 convexo tem uma frente de Pareto onde a superfície é quase plana na borda, enquanto a região intermediária é um pouco profunda, mas dando a característica de convexidade. Um dos desafios dos algoritmos neste problema é ser capaz de distribuir qualquer conjunto de partículas através da fronteira, evitando que eles permaneçam em apenas um lugar, como por exemplo, no meio da borda (DEB; JAIN, 2014a).

4.3 Indicadores de Qualidade

Na literatura, diferentes indicadores de qualidade são usados para avaliar os MOEAs (ADRA; FLEMING, 2011; SCHÜTZE; LARA; COELLO, 2011). Em nossos experimentos, já que estamos lidando com problemas de *benchmarking* onde a frente de Pareto pode ser obtida analiticamente, combinamos um conjunto de indicadores que medem a distância até a frente de Pareto. A Distância Geracional (GD) (COELLO; LAMONT; VELDHUIZEN, 2007) mede a distância entre o conjunto de aproximação gerado (PF_{approx}) e a verdadeira frente de Pareto do problema (PF_{true}). É uma medida de minimização e permite observar se o algoritmo converge para alguma região na frente de Pareto. A Distância Geracional Invertida (IGD) mede a distância mínima de cada ponto de PF_{true} aos pontos de PF_{approx} . IGD nos permite observar se PF_{approx} converge para frente de Pareto e também se este conjunto é bem diversificado.

Os indicadores de qualidade são comparados usando o teste de Kruskal-Wallis (ELLIOTT; HYNAN, 2011) a um nível de significância de 5%. Este teste é aplicado aos valores brutos de cada indicador. O teste de Kruskal-Wallis indica se há alguma diferença significativa entre cada conjunto de dados analisados e, em seguida, os valores médios e um procedimento pós-teste são usados para identificar qual algoritmo tem os melhores valores. A fim de auxiliar na análise dos resultados são utilizados gráficos *boxplots*. Um *boxplot* é um gráfico utilizado para descrever grupos de dados. Apresentando a mediana, os quartis inferiores e superiores, os valores limites do conjunto de dados e os possíveis *outliers* do conjunto. O teste de Kruskal-Wallis e o pós-teste

estão disponíveis no pacote estatístico R (TEAM et al., 2013). Destacando que cada algoritmo foi executado 20 vezes.

4.4 Algoritmos e Parâmetros

Essa seção é dividida em subseções, nas quais estão organizadas da seguinte forma: na subseção 4.4.1 é mostrada os parâmetros gerais, ou seja que são utilizados em todos os experimentos; na subseção 4.4.2 são apresentados os parâmetros do MSAT, no qual é o primeiro do conjunto de experimentos; na subseção 4.4.3 são mostrados os parâmetros específicos do MSAT-H e na subseção 4.4.4 são apresentados os parâmetros do MSAT-MH, nos quais fazem parte do segundo e terceiro conjunto de experimentos, respectivamente.

4.4.1 Parâmetros Gerais

Para medição do desempenho os resultados são comparados com o algoritmo NSGA-III. Aqui, é utilizado uma versão do NSGA-III disponível no *framework* jMetal (DURILLO; NEBRO, 2011). Os parâmetros do algoritmo NSGA-III foram definidos de acordo com (DEB; JAIN, 2014a), onde os parâmetros pc , pm , pr são respectivamente a probabilidade de combinação SBX, probabilidade de mutação Polinomial e a quantidade de pontos de referência. A relação desses parâmetros são mostrados a seguir:

Tabela 1 – Parâmetros do NSGA-III

	pc	pm	pr
3 obj	1	1/n	92
5 obj	1	1/n	212
10 obj	1	1/n	276

A fim de fazer uma comparação justa entre todos os algoritmos, cerca de 48000 chamadas de função de avaliação foram executadas para todos os algoritmos testados. Com a configuração exposta, a próxima etapa é comparar os parâmetros de cada experimento e os resultados das execuções para ver qual algoritmo mais adequado para cada tipo de problema. Os parâmetros do MSAT (adotado para todos os experimentos) são: 30 enxames, 160 partículas por enxame, 50 iterações para atualização e 100 soluções em cada arquivo, onde o número de enxames e o momento de comunicação foram definidos experimentalmente.

4.4.2 Parâmetros do Experimento I

O presente trabalho adotou a combinação entre diferentes tipos de arquivadores conhecidos na literatura. Dito isto, os arquivadores usados aqui são mostrados na Tabela 2.

Tabela 2 – Arquivadores

Métodos	Descrição
Ideal	Ideal Archiver (BRITTO; POZO, 2012a)
MGA	Multi-level Grid Archiving (LAUMANN; ZENKLUSEN, 2011)
Crowd	Crowding Distance Archiver (NEBRO et al., 2008)
SPEA2	SPEA2 Density Archiver (ZITZLER; LAUMANN; THIELE, 2002)

Qualquer tipo de arquivadores pode ser usado no MSAT. Aqui, nosso objetivo é explorar a combinação de diferentes arquivadores com o objetivo de encontrar aqueles que têm melhores resultados em MaOPs. Embora existam várias formas diferentes para combinar os métodos de arquivamento, este trabalho adotou a metodologia de usar pares desses métodos. Os arquivadores são distribuídos sobre os diferentes números de enxames, repetidamente, até que todos os enxames tenham sido atribuídos com um tipo de arquivadores.

Além disso, de acordo com ([BRITTO; POZO, 2012b](#)), Arquivos Ideal e MGA obtiveram melhores resultados quando aplicados a um algoritmo MOPSO testado em problemas DTLZ. Além disso, a combinação de Ideal e MGA em um algoritmo *multi-swarm* foi previamente explorada em ([BRITTO; MOSTAGHIM; POZO, 2014](#)) obtendo bons resultados.

Tabela 3 – Combinações entre os arquivadores

ID	Topologia	Combinação
1	Anel	Ideal e Crowd
2	Anel	Ideal e SPEA2
3	Anel	Ideal e MGA
4	Anel	MGA e Crowd
5	Anel	MGA e SPEA2
6	Grafo Completo	Ideal e Crowd
7	Grafo Completo	Ideal e SPEA2
8	Grafo Completo	Ideal e MGA
9	Grafo Completo	MGA e Crowd
10	Grafo Completo	MGA e SPEA2

Aqui, essa combinação é estendida considerando todos os arquivos descritos na Tabela 2, porém foi definido que pelo menos um desses dois arquivadores deve ser selecionado. Estas restrições geraram a combinação de pares dos arquivos explorados pelo MSAT, descritos na Tabela 3, onde também é apresentada a topologia de comunicação, gerando 10 configurações diferentes do MSAT.

4.4.3 Parâmetros do Experimento II

Os parâmetros de MSAT-H adotados foram as variações de 21, 45, 66 e 91 pontos de referência na intenção de investigar experimentalmente se a geração de poucos pontos iria trazer um bom comportamento, caracterizando assim várias configurações de MSAT-H e posteriormente verificando qual dessas se destacava.

4.4.4 Parâmetros do Experimento III

Os parâmetros de MSAT-MH como particularidade que são destacados são os tipos de funções de transformação que são aplicadas nos pontos de referência, onde provavelmente dependendo do tipo de função e do tipo de problema o algoritmo possa comportar-se de forma esperada e atingir convergência e diversidade em frente à fronteira. Foram aplicadas funções matemáticas nos parâmetros do MSAT-H, nos quais foram mostrados na seção anterior, onde foram adotados variações de 21, 45, 66 e 91 pontos de referência e aplicadas as funções descritas na seção 3.3 no intuito de identificar o melhor desempenho dentre esses parâmetros.

4.5 Experimento I

O experimento I apresenta o MSAT que serviu de base para a construção dos outros algoritmos, onde é um algoritmo baseado em arquivamento e topologias. Ainda nessa seção são mostrados os resultados obtidos pelo MSAT, nos quais, também foram apresentados na conferência do CEC 2017 (*IEEE Congress on Evolutionary Computation 2017*, conforme (MATOS; BRITTO, 2017)).

4.5.1 Análise de parâmetros do MSAT

Primeiro, as diferentes configurações do MSAT foram confrontadas. Tabela 4 mostra a análise resumida das melhores combinações de arquivadores. Cada célula apresenta a melhor

Tabela 4 – Análise resumida das melhores combinações de arquivadores

Problema	Indicador	3 obj	5 obj	10 obj
DTLZ2	GD	1, 2 e 3	1, 2, 3 e 4	1, 2 e 3
	IGD	1, 2 e 3	3, 4, 5 e 10	1, 2, 3 e 6
DTLZ4	GD	6, 7 e 8	1, 2 e 3	1, 2 e 3
	IGD	8, 9 e 10	1, 3, 4, 5, 8, 9 e 10	1, 2, 3 e 5
DTLZ6	GD	1, 2, 3, 4 e 5	1, 2, 3, 4, 5 e 10	1, 2 e 3
	IGD	1, 2, 3, 4 e 5	2, 5, 7 e 10	2, 5, 7 e 10
DTLZ2 Convexo	GD	1, 3, 6, 8, 9 e 10	1, 3, 4 e 5	1, 2, 3 e 5
	IGD	4, 5, 9 e 10	2, 5 e 10	4, 5 e 10

combinação (o ID é descrito na Tabela 3) do MSAT que obteve os melhores resultados de acordo com o pós-teste de Kruskal-Wallis (ELLIOTT; HYNAN, 2011), para GD e IGD, respectivamente,

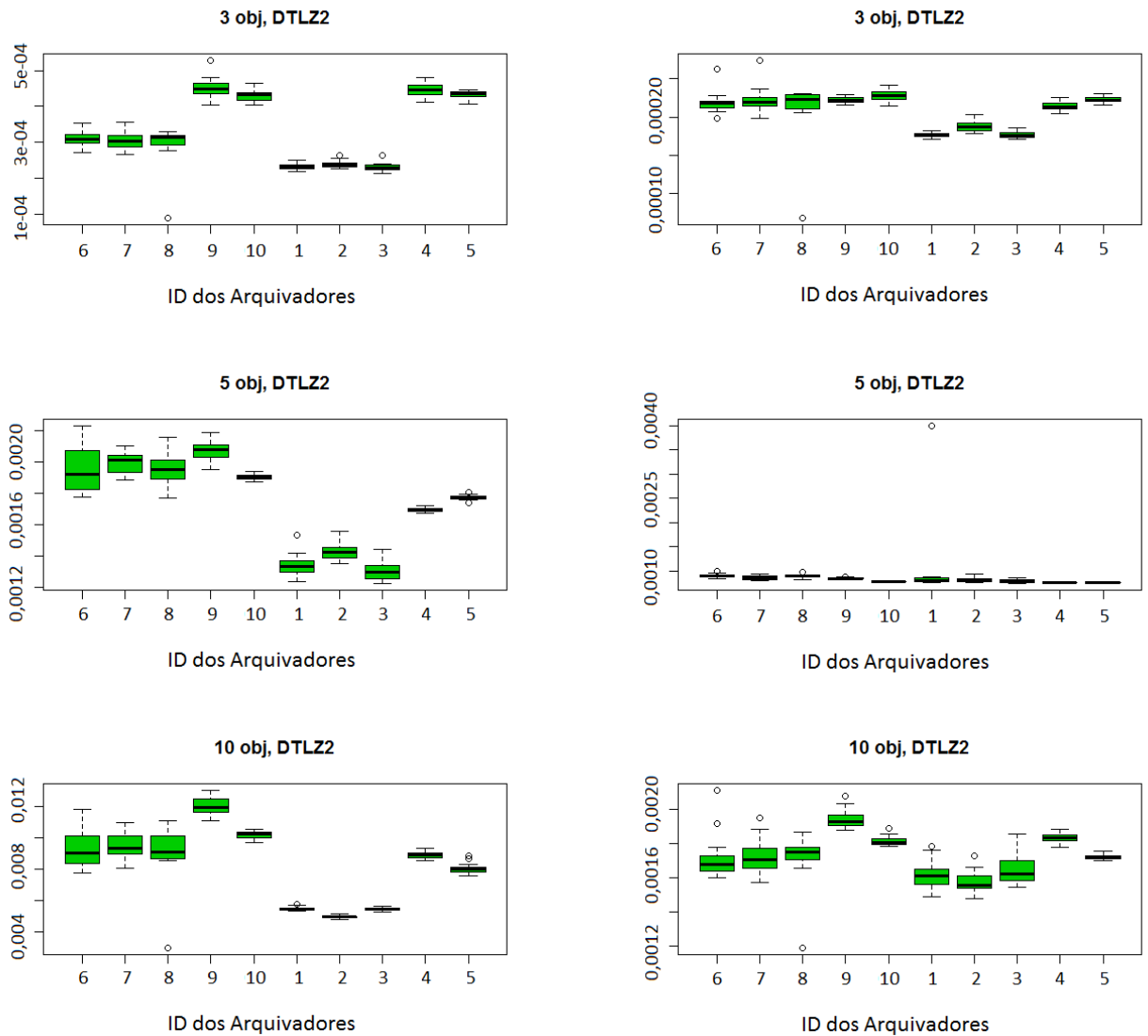


Figura 9 – *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2.

em cada número de objetivos. No intuito de facilitar na interpretação do teste de Kruskal-Wallis são apresentados alguns gráficos *boxplots*, Figuras 9, 10, 11 e 12.

DTLZ2 e DTLZ2 Convexo são problemas usados para medir a escalabilidade de um determinado MOEA. Com isso, ambos os problemas colocam obstáculos, dificultando, portanto, tanto em termos de convergência quanto na diversidade em relação à fronteira de Pareto (DEB; JAIN, 2014a). Para o problema do DTLZ2, as configurações que obtiveram bons resultados em termos de GD e IGD foram algoritmos de 1, 2 e 3 conforme pode-se observar na Figura 9 (todos os quais usam o arquivo Ideal). Além disso, o algoritmo 3 obteve os melhores resultados para todas as funções objetivo de acordo com o teste de Kruskal-Wallis. A topologia de anel pode ser destacada, uma vez que a grande maioria dos algoritmos com melhor GD e IGD usou essa topologia. A topologia de grafo completo só obteve bons resultados em algoritmos 6 e 10, mas algoritmos 7, 8 e 9 não se destacaram em nenhuma das funções objetivo. Em resumo, para uma

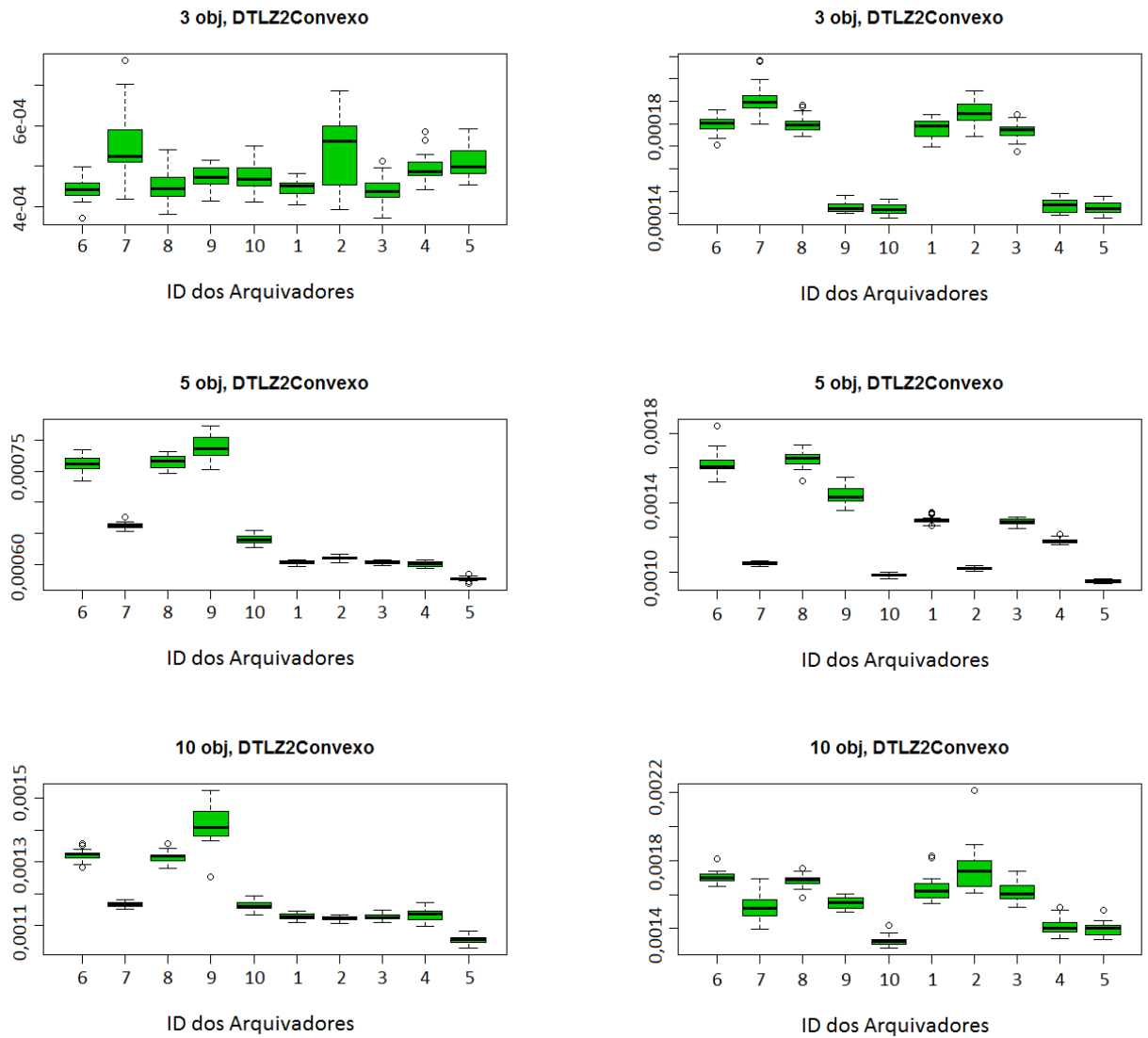


Figura 10 – *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.

fronteira de Pareto côncava, destacou-se o uso de um arquivador ideal em uma topologia de anel.

Para DTLZ2 Convexo, algoritmo 5 pode ser destacado. Ele aparece com relevância em quase todas as funções objetivo, tanto para GD quanto para IGD. Algoritmo 10 também teve bons resultados, especialmente em termos de IGD como pode-se observar na Figura 10. Isso indica que a combinação de arquivadores MGA e SPEA2 se destacou em um problema convexo, independentemente da topologia. Os resultados de 1 e 3 (ambos usam Ideal) estão entre os melhores na análise de GD. No entanto, eles não obtiveram bons resultados de IGD. Isso indica que o uso do Ideal pode levar o MSAT a convergir em um problema convexo, porém o algoritmo não terá uma boa diversidade. Algoritmos 6, 8 e 9 apareceram com bons resultados para um pequeno número de funções objetivo, mas o desempenho deteriorou-se quando o número de funções objetivo é aumentado.

O problema DTLZ4 dificulta a obtenção de diversidade em frente a fronteira de Pareto.

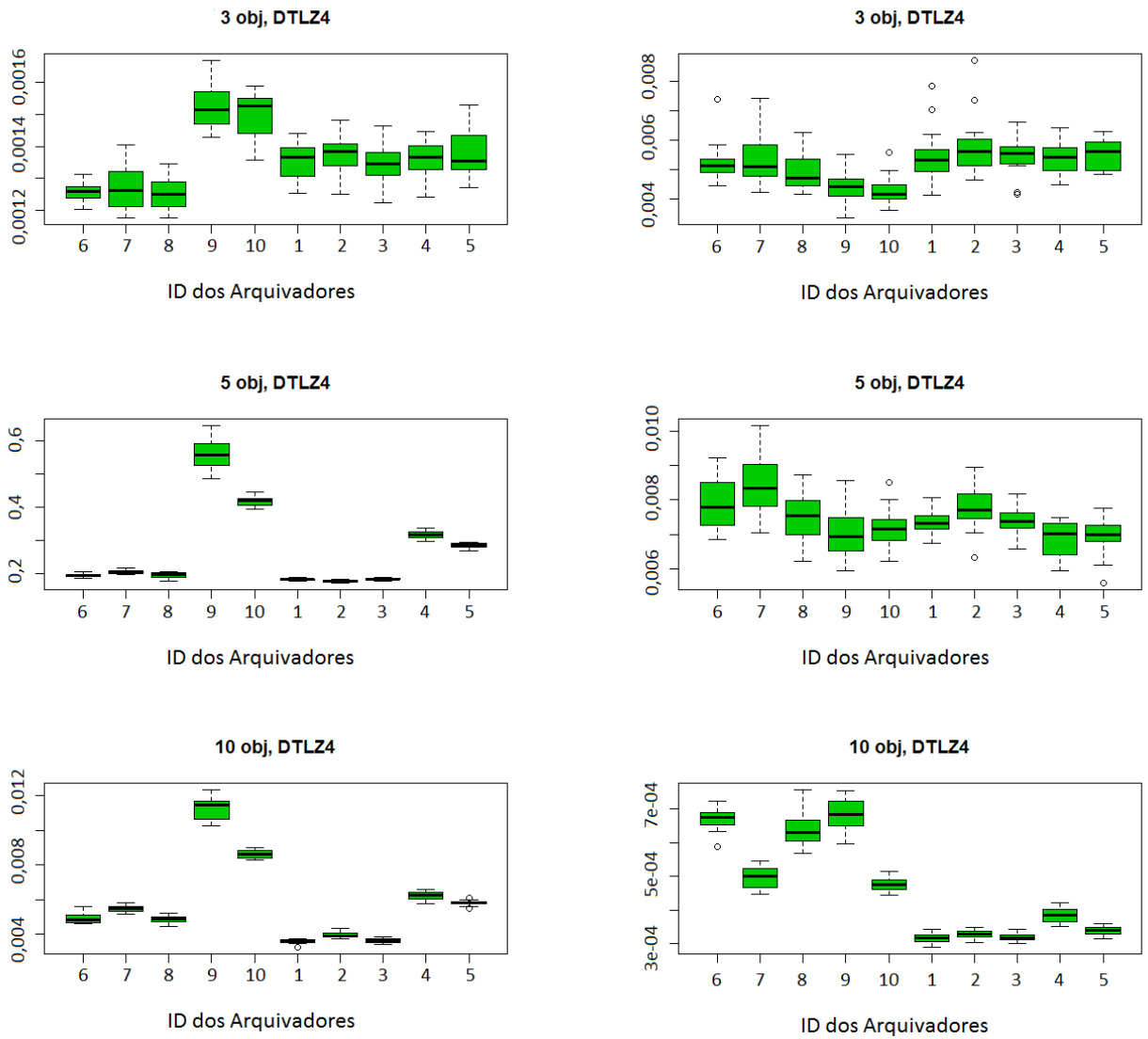


Figura 11 – *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ4.

Como desejamos observar o comportamento dos algoritmos em termos de diversidade, a medida IGD é destacada como a medida mais relevante. O MSAT que se destacou para maior dimensionalidade foi 1, 3 e 5, pois tiveram bons resultados no IGD como pode-se observar na Figura 11. Todos eles usam topologia de anel. Algoritmos 1 e 3 também tiveram resultados de GD significativos. Algoritmos 8, 9 e 10 tiveram bons resultados de IGD para 3 e 5 objetivos, mas quando o número de funções objetivo aumentou, eles não obtiveram resultados satisfatórios. Assim, a topologia de anel destacou-se para o problema DTLZ4. Também os arquivadores Ideal e MGA tiveram os melhores resultados.

DTLZ6 apresenta vários ótimos locais, o que torna a pesquisa mais difícil de convergir para a verdadeira fronteira de Pareto. Como desejamos observar o comportamento dos algoritmos em termos de convergência, a medida GD é destacada como a mais relevante. Assim, de acordo com os resultados apresentados na tabela 4, os algoritmos que tiveram valores relevantes em

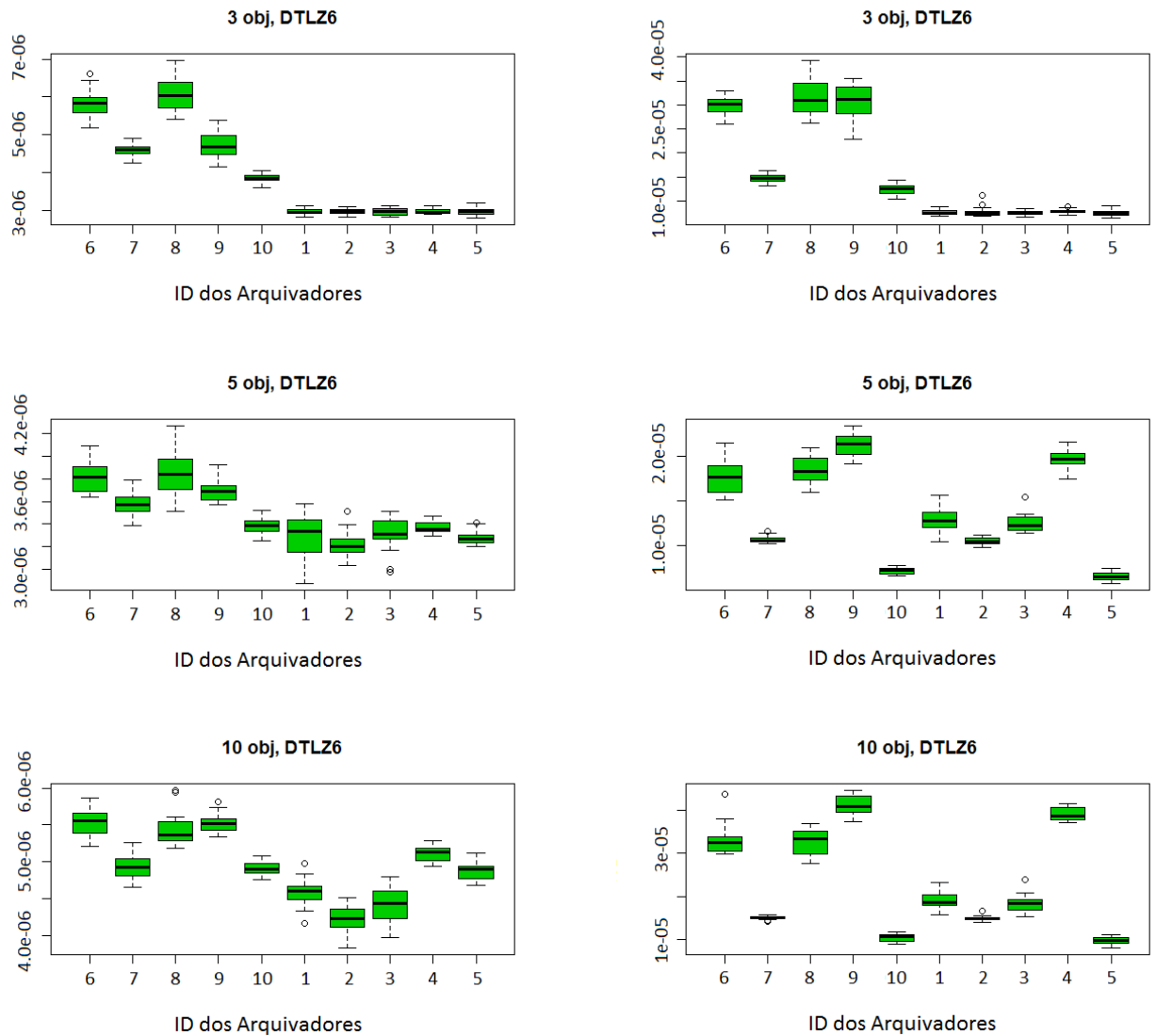


Figura 12 – *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ6.

GD em todo o número de funções objetivo foram 1, 2 e 3 como mostra a Figura 12. Além disso, o algoritmo 2 possui os melhores valores de IGD. Assim, podemos destacar que o uso do arquivador ideal com topologia de anel levou o MSAT a melhores resultados. No entanto, como DTZ2, o DTLZ6 tem uma forma côncava. O algoritmo 5 se destacou em quase todos os cenários, mas foi superado quando o problema possui 10 funções objetivo. Isso enfatiza a importância da topologia de anel para a convergência. Algoritmo 4 teve bom GD para 3 e 5 objetivos. Os outros algoritmos não se destacaram.

Em resumo, a topologia de anel teve melhores resultados do que a de grafo completo. Ela teve melhores resultados nos problemas DTLZ2, DTLZ4 e DTLZ6 para quase todos os cenários. Somente para o DTLZ2 Convexo, um problema de forma convexa, a topologia de grafo completo teve resultados similares a topologia de anel. No entanto, mesmo neste caso, ambos os algoritmos compartilhavam o mesmo conjunto de arquivadores, o que indica que esses arquivadores eram

responsáveis pelos bons resultados. Considerando a combinação de algoritmos, o Ideal destacou-se para o DTLZ2 e DTLZ4, independentemente das outras combinações de arquivadores. O arquivador MGA teve bons resultados para DTLZ4 e o DTLZ2 Convexo. Logo, a combinação de ambos deve ser considerada para obter bons resultados em termos de convergência ou quando se trata de problemas de forma côncava. O MGA deve ser considerado quando visa obter maior diversidade ou quando é utilizado em problema de forma convexa.

4.5.2 Resultados e Discussão

A seção anterior discutiu a influência dos arquivadores e topologias no MSAT aplicado aos MaOPs. Além dessa análise, é importante enfrentar o MSAT contra outro algoritmo estado-da-arte aplicado a otimização com muitos objetivos. Para este propósito, o algoritmo NSGA-III foi escolhido, uma vez que é especialmente designado para MaOPs. Além de ser frequentemente usado como algoritmo de base da literatura na temática de muitos objetivos, ele foi avaliado usando a família DTLZ em (DEB; JAIN, 2014a), obtendo bons resultados. Em (DEB; JAIN, 2014a), o NSGA-III superou algumas versões do MOEA/D nos problemas DTLZ2, DTLZ4 e DTLZ2 Convexo.

Tabela 5 – Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	Obj	ID 1	ID 2	ID 3	ID 5	ID 10	NSGIII
DTLZ2	3	2,32E-04 (6,90E-06)	2,39E-04 (9,35E-06)	2,32E-04 (1,04E-05)	4,32E-04 (9,91E-06)	4,30E-04 (1,53E-05)	1,12E-03 (3,75E-05)
	5	1,34E-03 (6,50E-05)	1,43E-03 (5,01E-05)	1,31E-03 (6,36E-05)	1,77E-03 (1,59E-05)	1,91E-03 (1,74E-05)	2,52E-03 (1,97E-06)
	10	5,48E-03 (1,09E-04)	4,96E-03 (8,58E-05)	5,47E-03 (9,67E-05)	8,04E-03 (3,11E-04)	1,02E-02 (2,21E-04)	1,72E-02 (7,02E-03)
DTLZ4	3	1,35E-03 (5,63E-05)	1,37E-03 (5,46E-05)	1,35E-03 (5,62E-05)	1,38E-03 (7,48E-05)	1,50E-03 (6,86E-05)	1,11E-03 (2,50E-05)
	5	1,81E-01 (2,84E-03)	1,76E-01 (3,11E-03)	1,82E-01 (3,29E-03)	2,84E-01 (7,33E-03)	4,17E-01 (1,36E-02)	2,52E-03 (1,24E-06)
	10	3,58E-03 (1,16E-04)	3,96E-03 (1,72E-04)	3,66E-03 (1,32E-04)	5,82E-03 (1,46E-04)	8,60E-03 (2,33E-04)	3,34E-02 (1,19E-04)
DTLZ6	3	2,96E-06 (8,35E-08)	2,97E-06 (6,90E-08)	2,98E-06 (9,84E-08)	2,96E-06 (9,67E-08)	3,85E-06 (1,09E-07)	1,09E-02 (1,10E-02)
	5	3,29E-06 (1,85E-07)	3,22E-06 (1,17E-07)	3,32E-06 (1,44E-07)	3,28E-06 (5,83E-08)	3,38E-06 (6,78E-08)	4,27E-02 (3,13E-03)
	10	4,58E-06 (1,84E-07)	4,21E-06 (1,68E-07)	4,42E-06 (2,38E-07)	4,87E-06 (1,19E-07)	4,91E-06 (8,72E-08)	3,58E-01 (3,29E-01)
DTLZ2 Convexo	3	4,47E-04 (1,77E-05)	5,37E-04 (8,07E-05)	4,42E-04 (3,08E-05)	5,12E-04 (3,72E-05)	4,73E-04 (3,44E-05)	8,35E-04 (4,73E-05)
	5	6,04E-04 (2,86E-06)	6,11E-04 (3,00E-06)	6,04E-04 (2,46E-06)	5,78E-04 (3,12E-06)	6,41E-04 (6,67E-06)	8,19E-04 (1,01E-06)
	10	1,13E-03 (1,09E-05)	1,12E-03 (6,44E-06)	1,13E-03 (8,99E-06)	1,05E-03 (1,13E-05)	1,16E-03 (1,49E-05)	3,60E-03 (2,75E-04)

Tabela 6 – Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	Obj	ID 1	ID 2	ID 3	ID 5	ID 10	NSGIII
DTLZ2	3	1,76E-04 (2,98E-06)	1,88E-04 (7,00E-06)	1,77E-04 (4,00E-06)	2,23E-04 (4,06E-06)	2,29E-04 (7,43E-06)	5,94E-04 (1,25E-06)
	5	9,67E-04 (6,96E-04)	8,22E-04 (4,06E-05)	7,96E-04 (3,24E-05)	7,60E-04 (6,37E-06)	7,87E-04 (7,06E-06)	8,60E-04 (1,60E-06)
	10	1,61E-03 (8,17E-05)	1,57E-03 (5,72E-05)	1,64E-03 (7,27E-05)	1,72E-03 (1,38E-05)	1,81E-03 (2,50E-05)	3,00E-03 (9,52E-05)
DTLZ4	3	5,44E-03 (8,32E-04)	5,73E-03 (9,22E-04)	5,42E-03 (6,42E-04)	5,53E-03 (4,70E-04)	4,30E-03 (4,62E-04)	1,07E-03 (1,42E-03)
	5	7,35E-03 (3,16E-04)	7,81E-03 (6,20E-04)	7,38E-03 (3,87E-04)	6,95E-03 (4,99E-04)	7,19E-03 (5,55E-04)	8,59E-04 (6,69E-07)
	10	3,15E-04 (1,31E-05)	3,28E-04 (1,30E-05)	3,19E-04 (1,26E-05)	3,40E-04 (1,26E-05)	4,75E-04 (1,96E-05)	9,02E-04 (8,35E-06)
DTLZ6	3	7,53E-06 (5,48E-07)	7,70E-06 (1,06E-06)	7,49E-06 (4,56E-07)	7,46E-06 (6,49E-07)	1,23E-05 (1,10E-06)	2,96E-03 (3,66E-05)
	5	1,28E-05 (1,23E-06)	1,05E-05 (3,87E-07)	1,26E-05 (9,45E-07)	6,56E-06 (4,88E-07)	7,16E-06 (3,36E-07)	5,38E-03 (1,73E-04)
	10	1,89E-05 (2,04E-06)	1,48E-05 (5,95E-07)	1,84E-05 (1,92E-06)	9,66E-06 (8,38E-07)	1,04E-05 (9,27E-07)	2,05E-02 (4,21E-03)
DTLZ2 Convexo	3	1,78E-04 (4,49E-06)	1,85E-04 (5,52E-06)	1,77E-04 (3,71E-06)	1,43E-04 (2,55E-06)	1,42E-04 (2,35E-06)	6,98E-04 (1,41E-05)
	5	1,30E-03 (2,02E-05)	1,02E-03 (8,00E-06)	1,29E-03 (1,78E-05)	9,44E-04 (8,55E-06)	9,80E-04 (9,85E-06)	1,21E-03 (5,43E-06)
	10	1,64E-03 (7,51E-05)	1,75E-03 (1,32E-04)	1,62E-03 (6,12E-05)	1,40E-03 (4,15E-05)	1,33E-03 (3,04E-05)	3,54E-03 (8,31E-04)

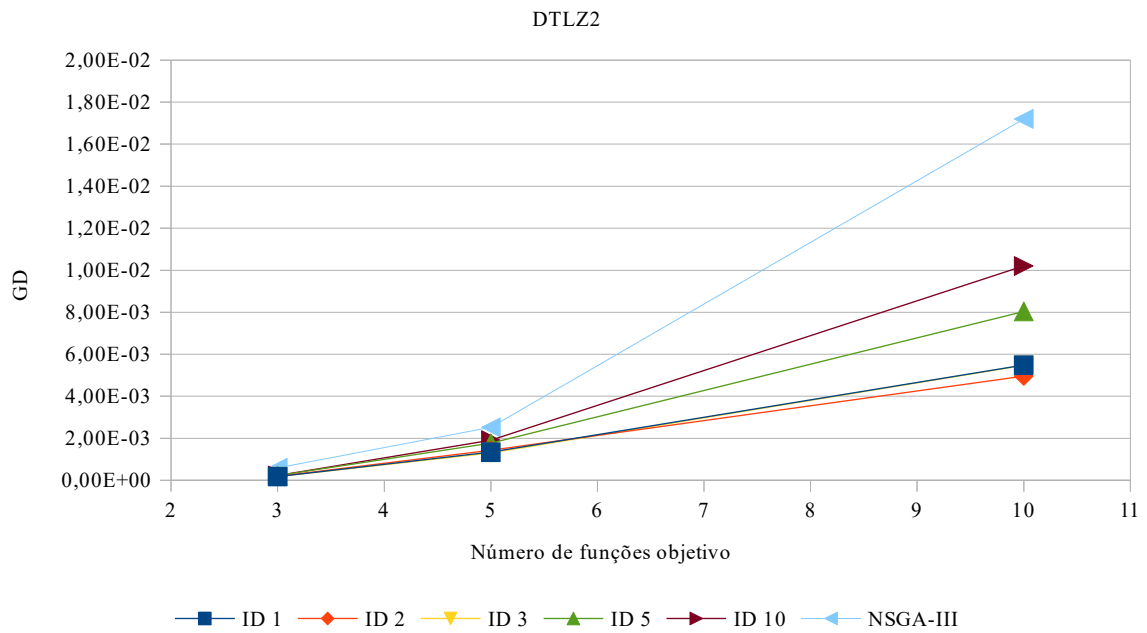


Figura 13 – Valores médios do indicador de qualidade GD para o problema DTLZ2.

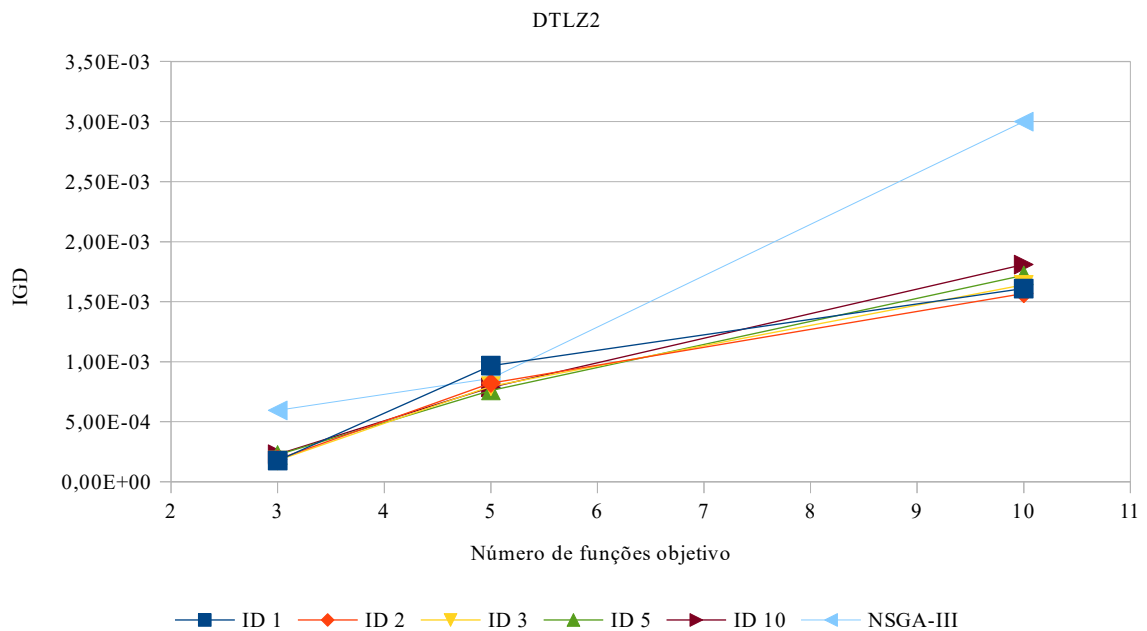


Figura 14 – Valores médios do indicador de qualidade IGD para o problema DTLZ2.

Uma vez que algumas configurações do MSAT se destacaram, apenas as melhores combinações de métodos de arquivamento e topologias são consideradas. Considerando a discussão da seção anterior, as configurações que usam topologia de anel, ID 1, 2, 3 e 5 foram escolhidas. Além disso, a configuração 10, que usa a topologia de grafo completo com MGA e SPEA2 também foi considerada. Essas configurações serão identificadas nesta seção como MSAT-1, MSAT-2, MSAT-3, MSAT-5 e MSAT-10.

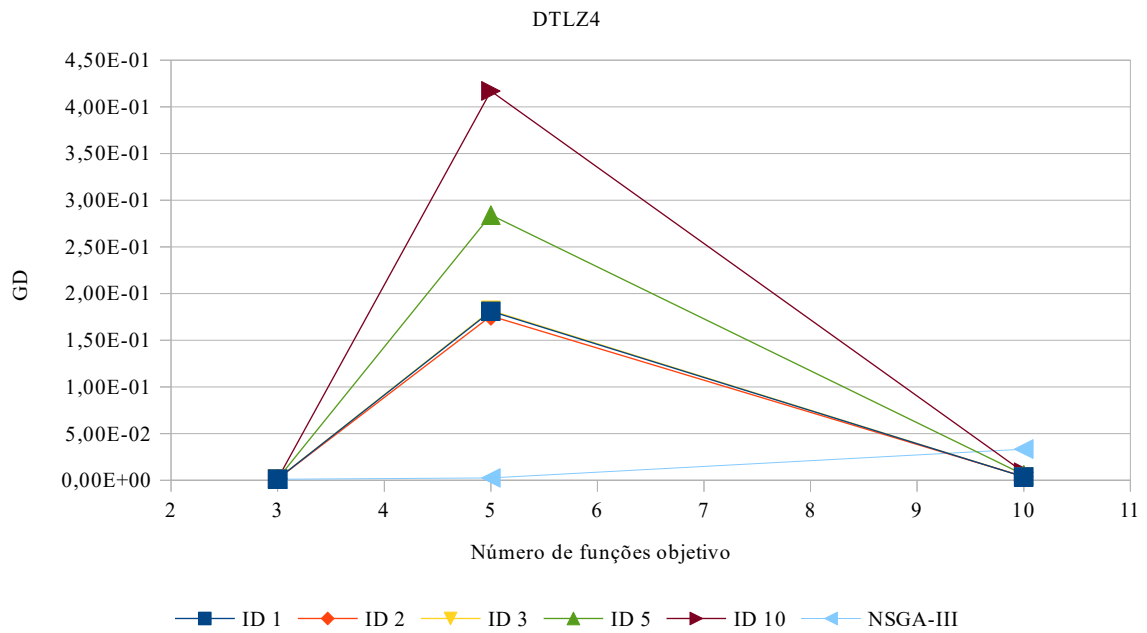


Figura 15 – Valores médios do indicador de qualidade GD para o problema DTLZ4.

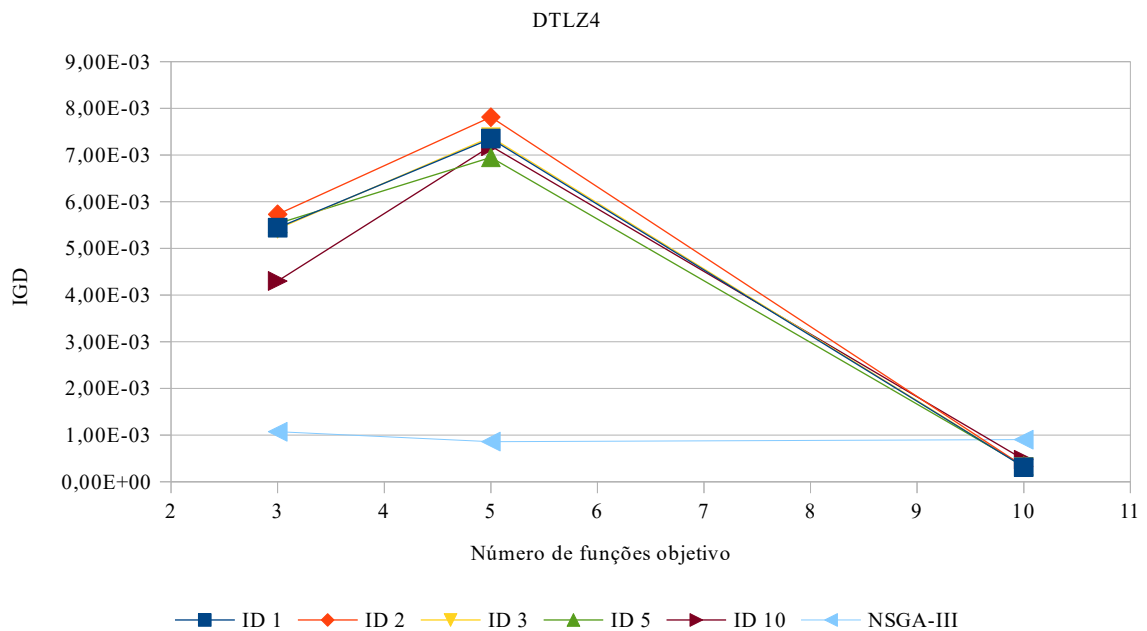


Figura 16 – Valores médios do indicador de qualidade IGD para o problema DTLZ4.

Os resultados são apresentados nas Tabelas 5 e 6. Diferente da análise anterior, essas tabelas apresentam os valores médios e o desvio padrão de GD e IGD obtidos por cada algoritmo. As células destacadas em cinza indicam os melhores resultados de acordo com o teste de Kruskal-Wallis. Além disso, as Figuras 13 até 20 apresentam os valores médios dos indicadores de qualidade. Cada curva no gráfico define como GD ou IGD mudou quando o número de funções objetivo aumenta, para cada algoritmo. Cada ponto representa o valor médio para um número específico de funções objetivo. É importante notar que a escala dos gráficos não é idêntica. A

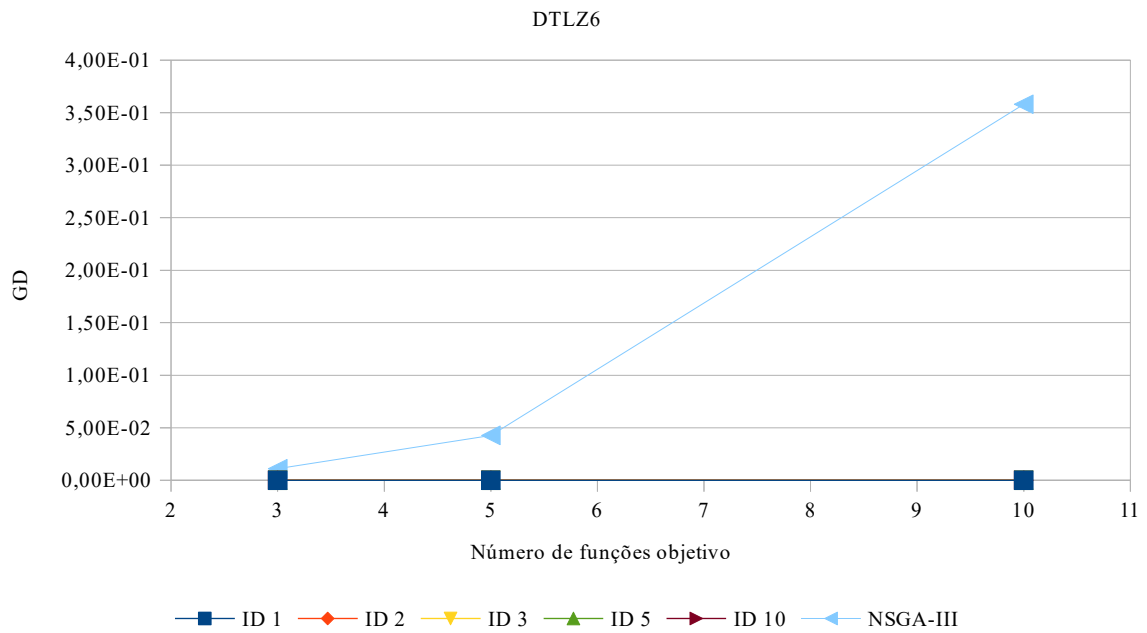


Figura 17 – Valores médios do indicador de qualidade GD para o problema DTLZ6.

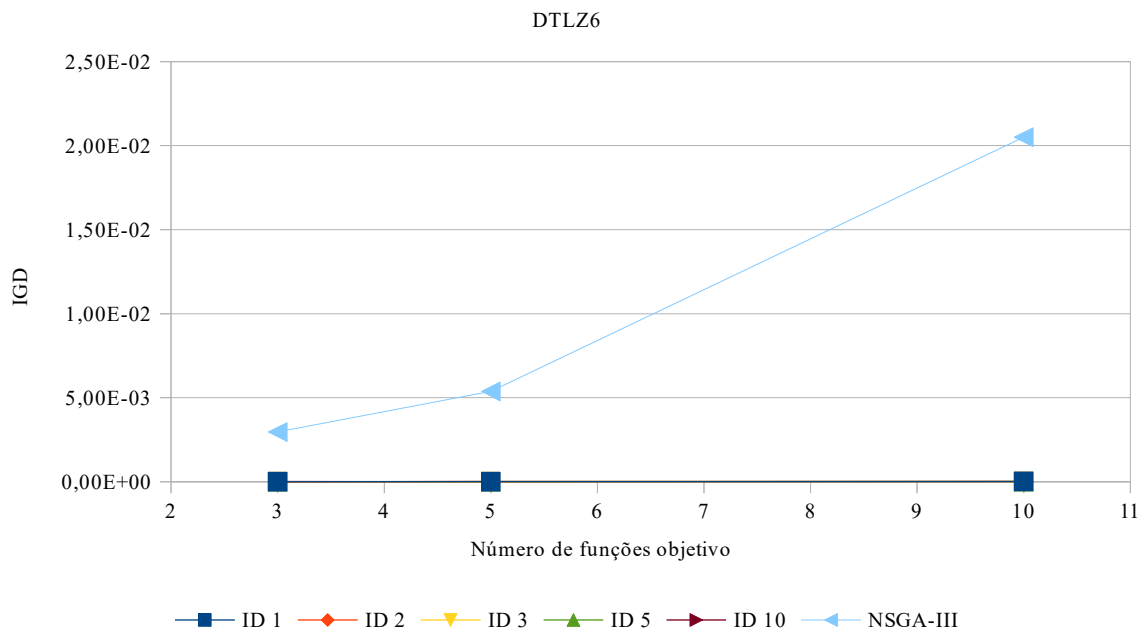


Figura 18 – Valores médios do indicador de qualidade IGD para o problema DTLZ6.

escala é definida individualmente para cada problema, de modo que as curvas possam mostrar as mudanças dos indicadores.

Para DTLZ2, o MSAT superou o NSGA-III, tanto no GD quanto no IGD. O MSAT com IDs 1, 2 e 3 obteve os melhores valores de GD e IGD em quase todos os cenários. O MSAT-5 obteve o melhor IGD para 5 funções objetivo. NSGA-III não se destacou em nenhum cenário. Estes bons resultados de MSAT podem ser observados nas Figuras 13 e 14. MSAT controlou

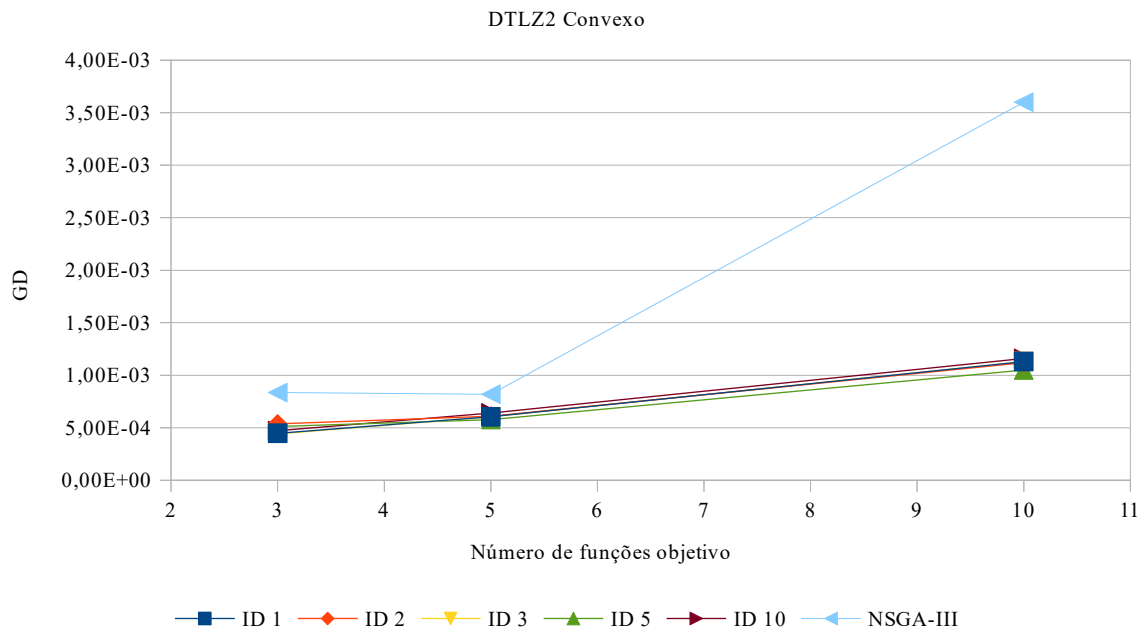


Figura 19 – Valores médios do indicador de qualidade GD para o problema DTLZ2 Convexo.

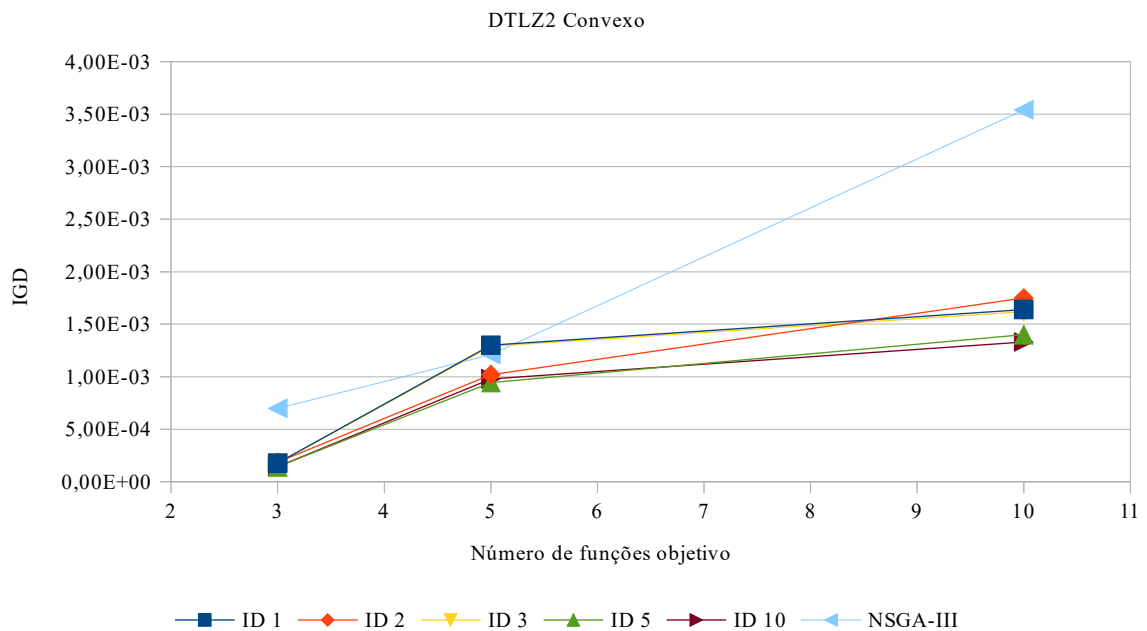


Figura 20 – Valores médios do indicador de qualidade IGD para o problema DTLZ2 Convexo.

melhor a deterioração quando o número de objetivos cresceu.

Os melhores resultados de MSAT sobre NSGA-III também foram observados no DTLZ2 Convexo. Novamente, o NSGA-III não obteve o melhor resultado em qualquer número de objetivos e foi superado pelo MSAT. Como este problema é de forma convexa, as configurações que usavam o MGA-SPEA2 se destacaram. Figuras 19 e 20 mostram os bons resultados do MSAT-5 e do MSAT-10. Além disso, podem-se observar os piores resultados de NSGA-III.

DTLZ4 foi o único problema em que o NSGA-III superou o MSAT. No entanto, no maior problema dimensional, com 10 funções objetivo, o MSAT teve os melhores resultados. A estratégia múltiplos enxames não funcionou bem para DTLZ4 com 3 e 5 funções objetivo. Na Figura 16, pode-se observar os piores valores de IGD obtidos pelo MSAT em comparação com NSGA-III. No entanto, a estratégia múltiplos enxames foi eficiente para 10 objetivos, e todas as configurações do MSAT obtiveram valores IGD melhores do que NSGA-III. Não é claro a partir desses resultados, por que o MSAT teve resultados ruins em 3 e 5 funções objetivo aplicado ao problema DTLZ4.

Finalmente, DTLZ6 apresentou a maior diferença entre MSAT e NSGA-III. O MSAT superou o NSGA-III para todas as funções objetivo em ambas as medidas. O MSAT obteve resultados muito bons de GD e pode controlar a deterioração em todos os objetivos analisados. Esse comportamento pode ser observado na Figura 17. NSGA-III piorou, enquanto o MSAT obteve quase o mesmo valor de GD.

Em resumo, podemos concluir que usar uma estratégia múltiplos enxames combinando arquivadores em diferentes topologias é uma boa maneira de resolver o MaOPS. O MSAT se destacou em relação ao NSGA-III para quase todos os cenários, considerando diferentes problemas e números de objetivos.

4.6 Experimento II

Como os resultados obtidos pelo MSAT mostraram que utilizar a topologia de anel junto aos arquivadores Ideal-MGA é interessante, foi adotado essa estratégia nesse segundo experimento e como peculiaridade do MSAT-H foi adicionada a característica de geração de pontos de referência dinamicamente. Foram gerados 21, 45, 66 e 91 pontos de referência na intenção de caracterizar várias configurações de MSAT-H e verificar qual dessas se destacava.

4.6.1 Análise de parâmetros do MSAT-H

As diferentes configurações do MSAT-H são confrontadas. As Tabelas 7 e 8 mostra a análise das configurações, onde cada célula em cinza apresenta o melhor resultado de MSAT-H. Nesse experimento foi percebido que a variação dos pontos fez que todas as configurações fossem todos iguais estatisticamente, com isso, foi adotado a escolha da configuração que obtiver menor média. No intuito de facilitar a interpretação são apresentados alguns gráficos *boxplots*, Figuras 21, 22, 23 e 24.

Tabela 7 – Quantidade de pontos de referência: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	6,7811E-03 (1,6697E-03)	6,7414E-03 (1,4005E-03)	6,9100E-03 (1,9128E-03)	6,6092E-03 (1,2591E-03)
	5	2,7093E-02 (1,1032E-03)	2,7782E-02 (9,2280E-04)	2,7285E-02 (8,6545E-04)	2,7425E-02 (1,0866E-03)
	10	2,2935E-02 (3,3881E-04)	2,2987E-02 (3,1426E-04)	2,3105E-02 (3,4129E-04)	2,2973E-02 (3,2108E-04)
DTLZ4	3	9,4880E-03 (1,2102E-03)	9,6613E-03 (1,0829E-03)	1,0052E-02 (1,3647E-03)	9,6346E-03 (1,5862E-03)
	5	2,7617E-02 (3,7039E-03)	2,6270E-02 (2,4401E-03)	2,6944E-02 (3,2863E-03)	2,6935E-02 (3,5278E-03)
	10	2,2688E-02 (6,0464E-04)	2,2473E-02 (5,7208E-04)	2,2589E-02 (6,6573E-04)	2,2824E-02 (9,6235E-04)
DTLZ6	3	6,9963E-01 (5,4366E-02)	7,0592E-01 (7,6801E-02)	7,0882E-01 (5,8048E-02)	6,9736E-01 (7,1945E-02)
	5	7,7667E-02 (5,9768E-03)	7,9757E-02 (4,7459E-03)	8,0949E-02 (4,7545E-03)	8,0949E-02 (5,7477E-03)
	10	2,7896E-02 (1,0061E-03)	2,8285E-02 (1,0313E-03)	2,7644E-02 (8,7697E-04)	2,8042E-02 (9,7239E-04)
DTLZ2 Convexo	3	9,3580E-02 (7,0478E-02)	6,6459E-02 (4,1357E-02)	9,4085E-02 (6,1701E-02)	1,0981E-01 (6,2516E-02)
	5	1,4217E-02 (7,8692E-03)	1,0901E-02 (2,3182E-03)	1,3192E-02 (9,4707E-03)	1,3919E-02 (6,5997E-03)
	10	9,1940E-03 (2,1689E-04)	9,1370E-03 (2,5949E-04)	9,1523E-03 (2,6352E-04)	9,1707E-03 (2,5395E-04)

Tabela 8 – Quantidade de pontos de referência: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	1,6187E-03 (1,4928E-04)	1,7035E-03 (1,4224E-04)	1,6984E-03 (1,7239E-04)	1,6421E-03 (1,5220E-04)
	5	3,9709E-03 (5,4506E-04)	3,7159E-03 (5,5125E-04)	3,5933E-03 (5,0864E-04)	3,5934E-03 (4,2001E-04)
	10	9,1500E-03 (2,1208E-04)	9,0894E-03 (2,7855E-04)	9,0708E-03 (2,4959E-04)	9,1373E-03 (2,3571E-04)
DTLZ4	3	1,2649E-02 (1,6413E-03)	1,3753E-02 (1,9237E-03)	1,2143E-02 (1,5164E-03)	1,1566E-02 (1,5135E-03)
	5	1,7288E-02 (2,2625E-03)	1,7712E-02 (1,6111E-03)	1,7066E-02 (1,3976E-03)	1,7280E-02 (1,9330E-03)
	10	4,3552E-02 (2,7452E-03)	4,4035E-02 (1,8098E-03)	4,3106E-02 (2,1066E-03)	4,3381E-02 (2,0400E-03)
DTLZ6	3	2,6215E-02 (4,4512E-03)	2,4330E-02 (5,2894E-03)	2,5372E-02 (3,9153E-03)	2,6484E-02 (4,0654E-03)
	5	2,3829E-03 (3,0041E-04)	2,5561E-03 (2,8417E-04)	2,4623E-03 (2,4473E-04)	2,5885E-03 (2,6986E-04)
	10	8,4655E-03 (5,5346E-04)	8,3530E-03 (7,8733E-04)	8,6068E-03 (8,5197E-04)	8,4460E-03 (8,2560E-04)
DTLZ2 Convexo	3	1,0932E-03 (2,0286E-04)	1,1602E-03 (1,8918E-04)	1,1636E-03 (2,5261E-04)	1,2234E-03 (2,9191E-04)
	5	5,6711E-03 (1,3252E-03)	5,7217E-03 (1,1523E-03)	6,0092E-03 (8,7061E-04)	5,9640E-03 (1,0368E-03)
	10	8,4026E-03 (1,0726E-04)	8,3028E-03 (3,4019E-04)	8,3973E-03 (6,5189E-05)	8,2619E-03 (3,0649E-04)

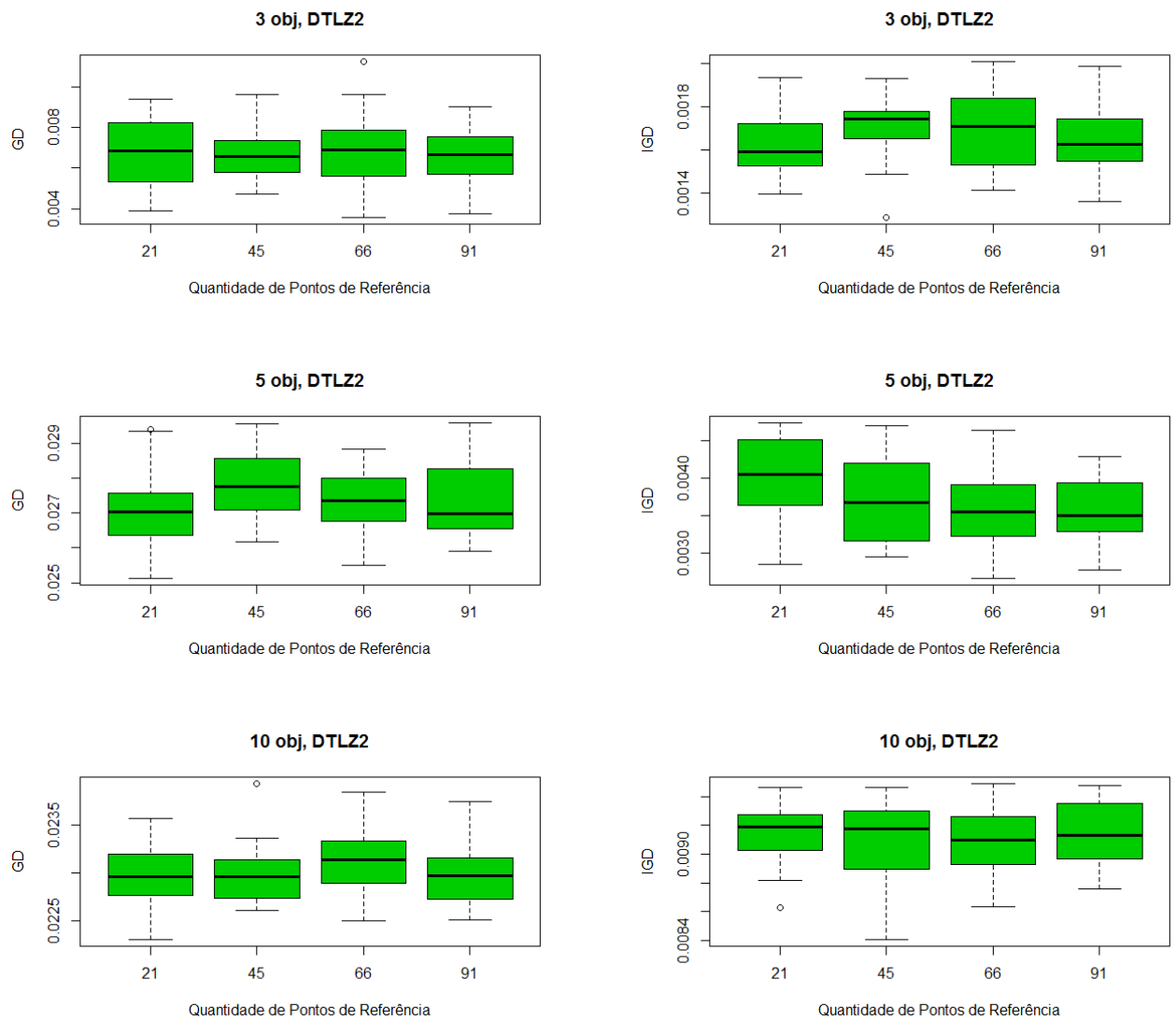


Figura 21 – Quantidade de pontos de referência: *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2.

Para o problema do DTLZ2, as configurações que obtiveram bons resultados em termos de GD foram algoritmos com 21 e 91 pontos conforme pode-se observar na Figura 21. Enquanto em termos de IGD as configurações de 21 e 66 pontos se destacaram. O algoritmo com 21 pontos obteve os melhores resultados para o problema DTLZ2 no indicador GD quando aplicado a 5 e 10 funções objetivo, já em IGD a configuração de 66 pontos obteve esse mesmo comportamento.

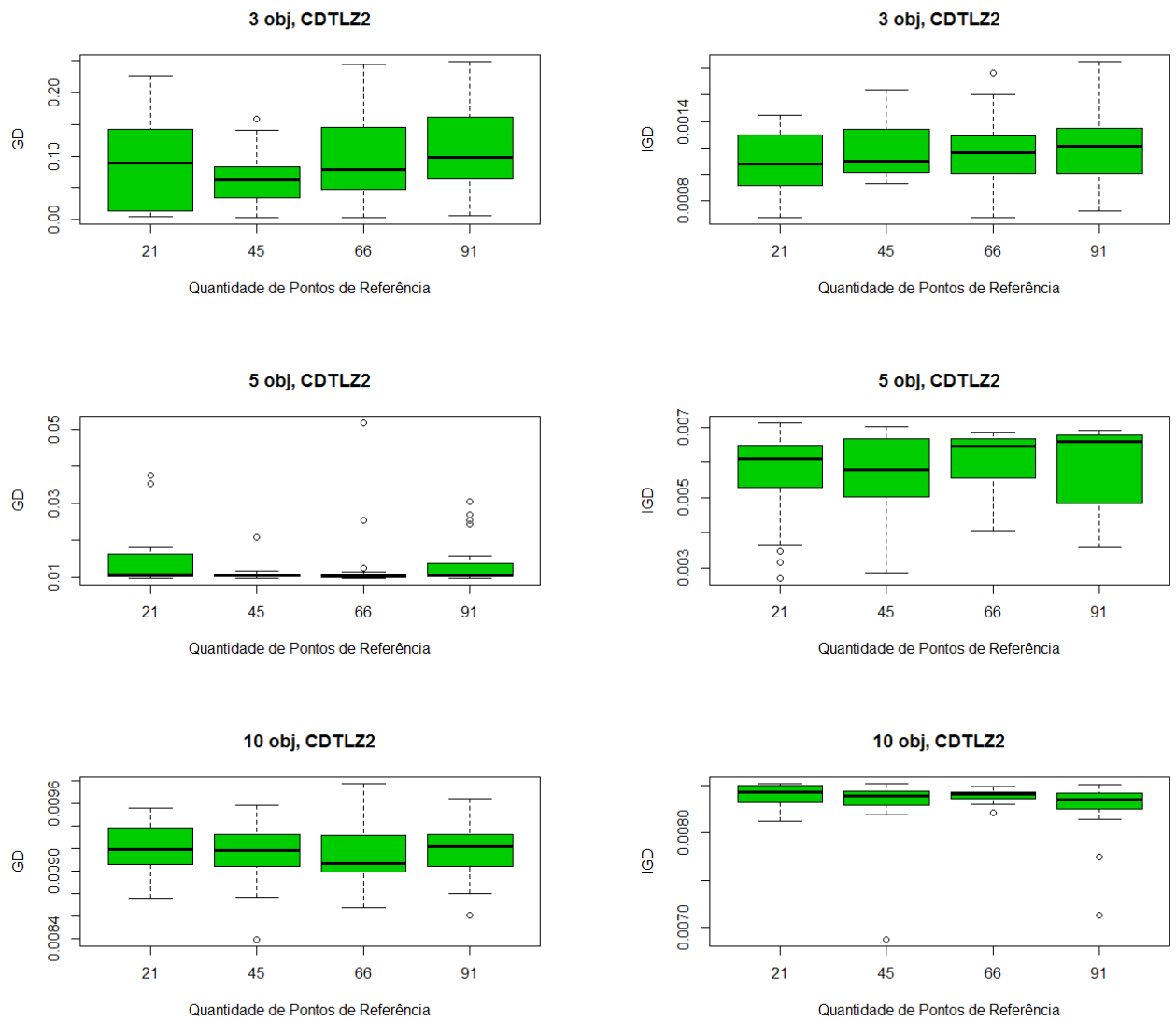


Figura 22 – Quantidade de pontos de referência: *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.

Para DTLZ2 Convexo, o algoritmo com 45 pontos pode ser destacado. Ele aparece com relevância em todas as funções objetivo no indicador GD, porém em IGD não obteve o mesmo êxito. O algoritmo com 21 e 91 pontos também conseguiu bons resultados, especialmente em termos de IGD como pode-se observar na Figura 22. A configuração de 21 pontos apareceu com bons resultados para um pequeno número de funções objetivo, mas o desempenho deteriorou-se quando o número de funções objetivo é aumentado.

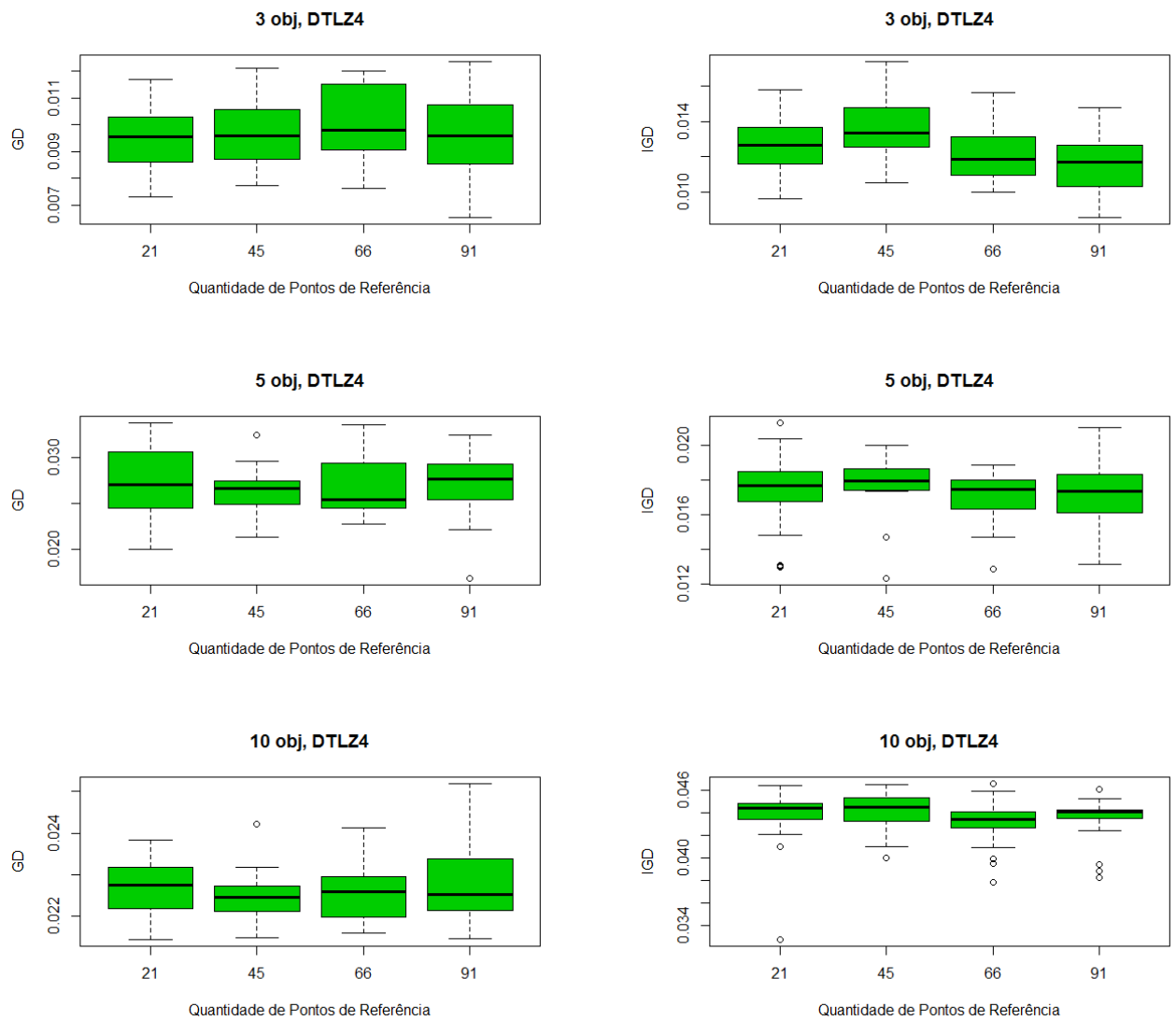


Figura 23 – Quantidade de pontos de referência: *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ4.

Como o problema DTLZ4 dificulta a obtenção de diversidade em frente a fronteira de Pareto e desejamos observar o comportamento dos algoritmos em termos de diversidade, a medida IGD é destacada como a medida mais relevante. O MSAT-H que se destacou para maior dimensionalidade foi a configuração de 66 pontos de referência, pois conseguiu bons resultados no IGD como pode-se observar na Figura 23. Algoritmos com 45 e 21 pontos também tiveram resultados de GD significativos. Algoritmo com 91 pontos conseguiu bons resultados de IGD para 3 objetivos, mas quando o número de funções objetivo aumentou, ele não obteve resultados satisfatórios.

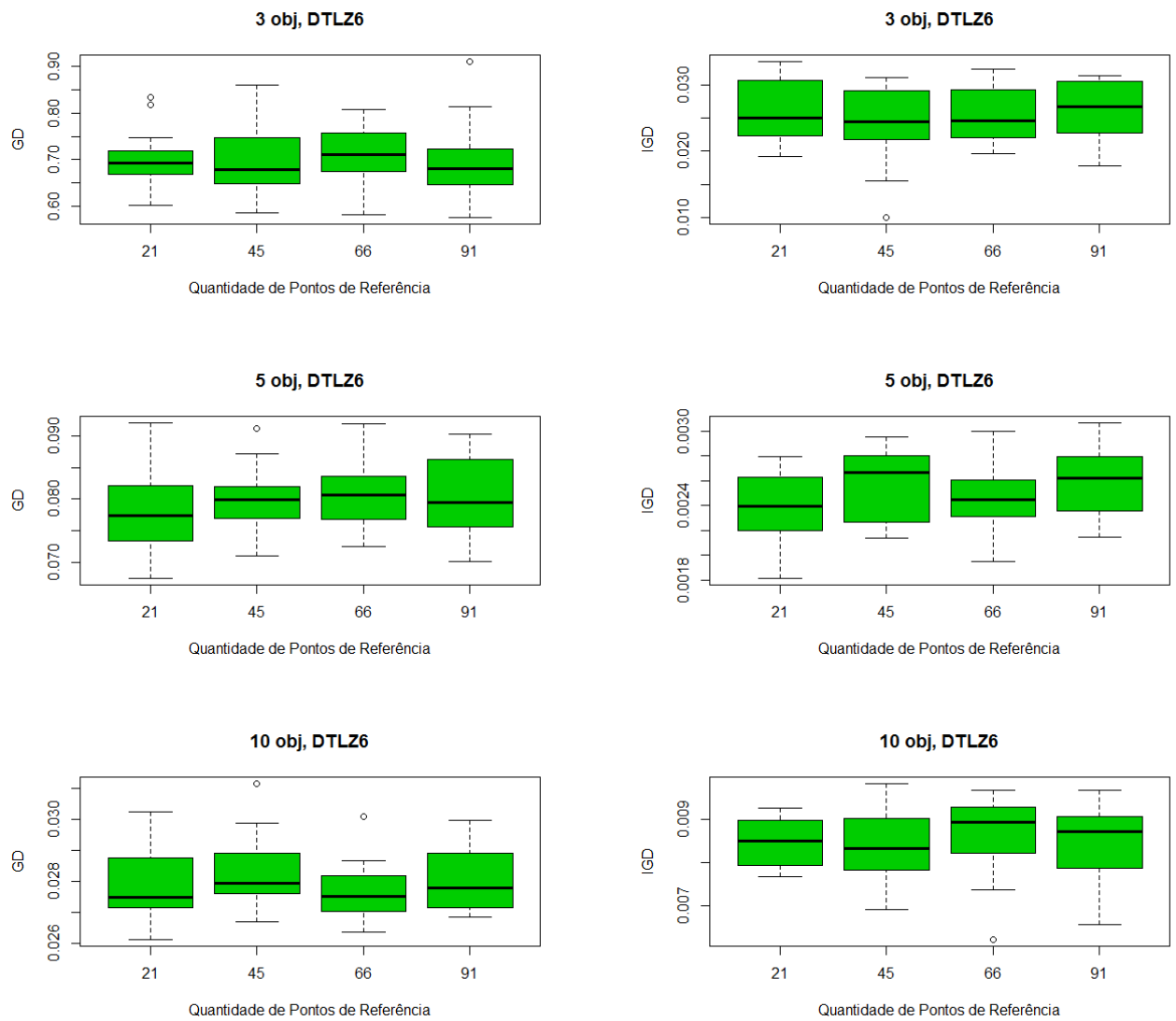


Figura 24 – Quantidade de pontos de referência: *Boxplot* gerado pelo pacote estatístico R dos valores GD e IGD com 3, 5 e 10 objetivos para o problema DTLZ6.

Como o problema DTLZ6 apresenta vários ótimos locais, isso torna a pesquisa mais difícil de convergir para a verdadeira fronteira de Pareto e como desejamos observar o comportamento dos algoritmos em termos de convergência, a medida GD é destacada como a mais relevante. Os algoritmos que tiveram valores relevantes em GD em todo o número de funções objetivo foram as configurações de 21, 66 e 91 pontos de referência como mostra a Figura 24. Além disso, os algoritmos com 21 e 45 pontos possui os melhores valores de IGD.

Em resumo, o algoritmo com 21 pontos de referência geralmente apareceu em evidência na maioria do problemas. Devido a esse resultado, foi escolhida essa configuração para ser comparada com o algoritmo MSAT e NSGA-III.

4.6.2 Resultados e Discussão

Primeiro, as diferentes configurações do MSAT-H foram confrontadas na seção anterior. As tabelas 9 e 10, diferente da análise anterior apresentam os valores médios e o desvio padrão de GD e IGD das melhores configurações de MSAT e MSAT-H que obteve os melhores resultados de acordo com o pós-teste de Kruskal-Wallis, para GD e IGD, respectivamente, em cada número de objetivos.

Tabela 9 – Comparação MSAT, MSAT-H e NSGA-III para o indicador GD.

Problema	obj	MSAT	MSAT-H	NSGAIII
DTLZ2	3	2,32E-04 (1,04E-05)	6,7811E-03 (1,6697E-03)	1,12E-03 (3,75E-05)
	5	1,31E-03 (6,36E-05)	2,7093E-02 (1,1032E-03)	2,52E-03 (1,97E-06)
	10	5,47E-03 (9,67E-05)	2,2935E-02 (3,3881E-04)	1,72E-02 (7,02E-03)
DTLZ4	3	1,35E-03 (5,62E-05)	9,4880E-03 (1,2102E-03)	1,11E-03 (2,50E-05)
	5	1,82E-01 (3,29E-03)	2,7617E-02 (3,7039E-03)	2,52E-03 (1,24E-06)
	10	3,66E-03 (1,32E-04)	2,2688E-02 (6,0464E-04)	3,34E-02 (1,19E-04)
DTLZ6	3	2,98E-06 (9,84E-08)	6,9963E-01 (5,4366E-02)	1,09E-02 (1,10E-02)
	5	3,32E-06 (1,44E-07)	7,7667E-02 (5,9768E-03)	4,27E-02 (3,13E-03)
	10	4,42E-06 (2,38E-07)	2,7896E-02 (1,0061E-03)	3,58E-01 (3,29E-01)
DTLZ2 Convexo	3	4,42E-04 (3,08E-05)	9,3580E-02 (7,0478E-02)	8,35E-04 (4,73E-05)
	5	6,04E-04 (2,46E-06)	1,4217E-02 (7,8692E-03)	8,19E-04 (1,01E-06)
	10	1,13E-03 (8,99E-06)	9,1940E-03 (2,1689E-04)	3,60E-03 (2,75E-04)

Tabela 10 – Comparação MSAT, MSAT-H e NSGA-III para o indicador IGD.

Problema	obj	MSAT	MSAT-H	NSGAIII
DTLZ2	3	1,77E-04 (4,00E-06)	1,6187E-03 (1,4928E-04)	5,94E-04 (1,25E-06)
	5	7,96E-04 (3,24E-05)	3,9709E-03 (5,4506E-04)	8,60E-04 (1,60E-06)
	10	1,64E-03 (7,27E-05)	9,1500E-03 (2,1208E-04)	3,00E-03 (9,52E-05)
DTLZ4	3	5,42E-03 (6,42E-04)	1,2649E-02 (1,6413E-03)	1,07E-03 (1,42E-03)
	5	7,38E-03 (3,87E-04)	1,7288E-02 (2,2625E-03)	8,59E-04 (6,69E-07)
	10	3,19E-04 (1,26E-05)	4,3552E-02 (2,7452E-03)	9,02E-04 (8,35E-06)
DTLZ6	3	7,49E-06 (4,56E-07)	2,6215E-02 (4,4512E-03)	2,96E-03 (3,66E-05)
	5	1,26E-05 (9,45E-07)	2,3829E-03 (3,0041E-04)	5,38E-03 (1,73E-04)
	10	1,84E-05 (1,92E-06)	8,4655E-03 (5,5346E-04)	2,05E-02 (4,21E-03)
DTLZ2 Convexo	3	1,77E-04 (3,71E-06)	1,0932E-03 (2,0286E-04)	6,98E-04 (1,41E-05)
	5	1,29E-03 (1,78E-05)	5,6711E-03 (1,3252E-03)	1,21E-03 (5,43E-06)
	10	1,62E-03 (6,12E-05)	8,4026E-03 (1,0726E-04)	3,54E-03 (8,31E-04)

Para DTLZ2, o MSAT-H não superou o NSGA-III e nem o MSAT, tanto no GD quanto no IGD. O MSAT-H com quantidade de 21 pontos de referência não obteve bons resultados, enquanto os melhores valores de GD e IGD em todos os cenários foram do MSAT. O MSAT se destacou na maioria das situações e por isso controlou melhor a deterioração na comparação com os outros algoritmos. DTLZ2 Convexo foi outro problema em que o MSAT superou o MSAT-H e o NSGA-III. No entanto, em problemas com menor dimensionalidade, com 3 funções objetivo, o NSGA-III teve os melhores resultados no indicador GD e em termos de IGD para 5 funções objetivo. Os resultados ruins de MSAT-H e MSAT sobre NSGA-III foram observados no DTLZ4. O NSGA-III obteve o melhor resultado em qualquer número de objetivos e superou o MSAT-H em todas as configurações, perdendo apenas para o MSAT quando aplicado a 10 funções objetivo.

O DTLZ6 apresentou a maior diferença entre MSAT, MSAT-H e NSGA-III, onde os melhores resultados de MSAT sobre o MSAT-H e NSGA-III foram observados neste problema, no qual o MSAT superou os outros dois para todas as medidas.

Em resumo, podemos concluir que a estratégia de variação de pontos de referência não obteve resultados satisfatórios quando comparados a esses algoritmos e aplicado a muitos objetivos. O MSAT se destacou em relação ao MSAT-H e ao NSGA-III para quase todos os cenários.

4.7 Experimento III

Os resultados obtidos pelo MSAT-H mostraram que utilizar somente a estratégia de variação de pontos de referência não obteve resultados satisfatórios. Com isso, foi realizado experimentos com interesse de verificar se essas configurações destacam-se quando modificado o hiperplano, logo, foi adotado essa estratégia nesse terceiro experimento.

4.7.1 Análise de parâmetros do MSAT-MH

As diferentes configurações do MSAT-MH são confrontadas. As Tabelas 11, 12, 13, 14, 15 e 16 mostram as análises das configurações, onde cada célula em cinza apresenta o melhor resultado de MSAT-MH. Nesse experimento foi percebido que a variação dos pontos com a aplicação das funções fizeram que todas as configurações fossem todas iguais estatisticamente, com isso, foi adotado a escolha da configuração que obtiver menor média. No intuito de facilitar a interpretação são apresentados alguns gráficos *boxplots*.

Tabela 11 – Função côncava: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	6,7273E-03 (1,4763E-03)	6,7921E-03 (1,3371E-03)	6,3493E-03 (1,4353E-03)	6,9173E-03 (1,0827E-03)
	5	6,2096E-03 (6,9515E-04)	6,1810E-03 (5,8794E-04)	6,4126E-03 (8,6050E-04)	6,5156E-03 (7,3748E-04)
	10	1,6657E-03 (9,8017E-05)	1,6841E-03 (9,9307E-05)	1,7244E-03 (9,8527E-05)	1,6988E-03 (8,9330E-05)
DTLZ4	3	9,6973E-03 (1,3740E-03)	9,2128E-03 (1,5083E-03)	9,4139E-03 (1,4525E-03)	9,5781E-03 (1,3337E-03)
	5	1,3675E+00 (1,3546E-01)	1,3069E+00 (1,1078E-01)	1,3024E+00 (8,8536E-02)	1,2895E+00 (1,5056E-01)
	10	1,7689E-03 (1,7583E-04)	1,7482E-03 (1,7217E-04)	1,7783E-03 (1,3471E-04)	1,7689E-03 (1,7583E-04)
DTLZ6	3	6,8797E-01 (5,5981E-02)	7,0770E-01 (4,9296E-02)	6,9539E-01 (3,9882E-02)	6,9514E-01 (6,6295E-02)
	5	2,2764E-01 (1,1206E-02)	2,3136E-01 (1,2066E-02)	2,3311E-01 (1,4852E-02)	2,2716E-01 (1,0381E-02)
	10	4,2188E-02 (2,6926E-03)	4,1824E-02 (2,9019E-03)	4,4690E-02 (5,6662E-03)	4,1359E-02 (2,1065E-03)
DTLZ2 Convexo	3	8,4076E-02 (5,4313E-02)	9,5028E-02 (6,4404E-02)	1,1501E-01 (4,9481E-02)	1,2195E-01 (6,6512E-02)
	5	1,1445E-02 (1,2969E-02)	1,6601E-02 (1,8466E-02)	1,7533E-02 (2,1578E-02)	1,4122E-02 (1,6784E-02)
	10	1,2306E-03 (5,1828E-04)	1,6178E-03 (2,1805E-03)	1,3557E-03 (1,3658E-03)	1,2670E-03 (4,2978E-04)

Tabela 12 – Função côncava: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	1,6428E-03 (1,5985E-04)	1,7381E-03 (1,3499E-04)	1,7115E-03 (1,5616E-04)	1,5792E-03 (1,5786E-04)
	5	3,6311E-03 (3,1341E-04)	3,5593E-03 (3,6973E-04)	3,6517E-03 (3,1104E-04)	3,4117E-03 (2,7126E-04)
	10	5,8229E-03 (6,1381E-04)	5,9073E-03 (6,1374E-04)	5,9412E-03 (6,0108E-04)	6,0427E-03 (5,4528E-04)
DTLZ4	3	1,1783E-02 (1,5938E-03)	1,2630E-02 (2,3159E-03)	1,2355E-02 (1,3856E-03)	1,1844E-02 (1,9631E-03)
	5	1,4015E-02 (1,0555E-03)	1,3668E-02 (7,8541E-04)	1,3788E-02 (8,6606E-04)	1,4001E-02 (8,1999E-04)
	10	1,3077E-02 (2,2870E-03)	1,3608E-02 (4,2273E-04)	1,2546E-02 (3,1108E-03)	1,2864E-02 (2,1120E-03)
DTLZ6	3	2,6705E-02 (4,6261E-03)	2,6198E-02 (4,8070E-03)	2,7063E-02 (3,1329E-03)	2,6954E-02 (3,5536E-03)
	5	2,2914E-02 (3,2316E-03)	2,0439E-02 (3,4283E-03)	2,0590E-02 (3,7912E-03)	1,9417E-02 (5,1733E-03)
	10	1,5963E-04 (5,1016E-05)	1,7379E-04 (9,8004E-05)	1,9305E-04 (8,0416E-05)	1,7072E-04 (4,5838E-05)
DTLZ2 Convexo	3	1,0959E-03 (2,6308E-04)	1,1295E-03 (2,0319E-04)	1,1678E-03 (2,3104E-04)	1,2094E-03 (2,1631E-04)
	5	1,5992E-02 (2,9112E-03)	1,6955E-02 (3,1385E-03)	1,6963E-02 (2,2880E-03)	1,6534E-02 (3,0751E-03)
	10	1,4093E-02 (2,6308E-03)	1,4593E-02 (2,9589E-03)	1,4334E-02 (2,2657E-03)	1,3539E-02 (3,0782E-03)

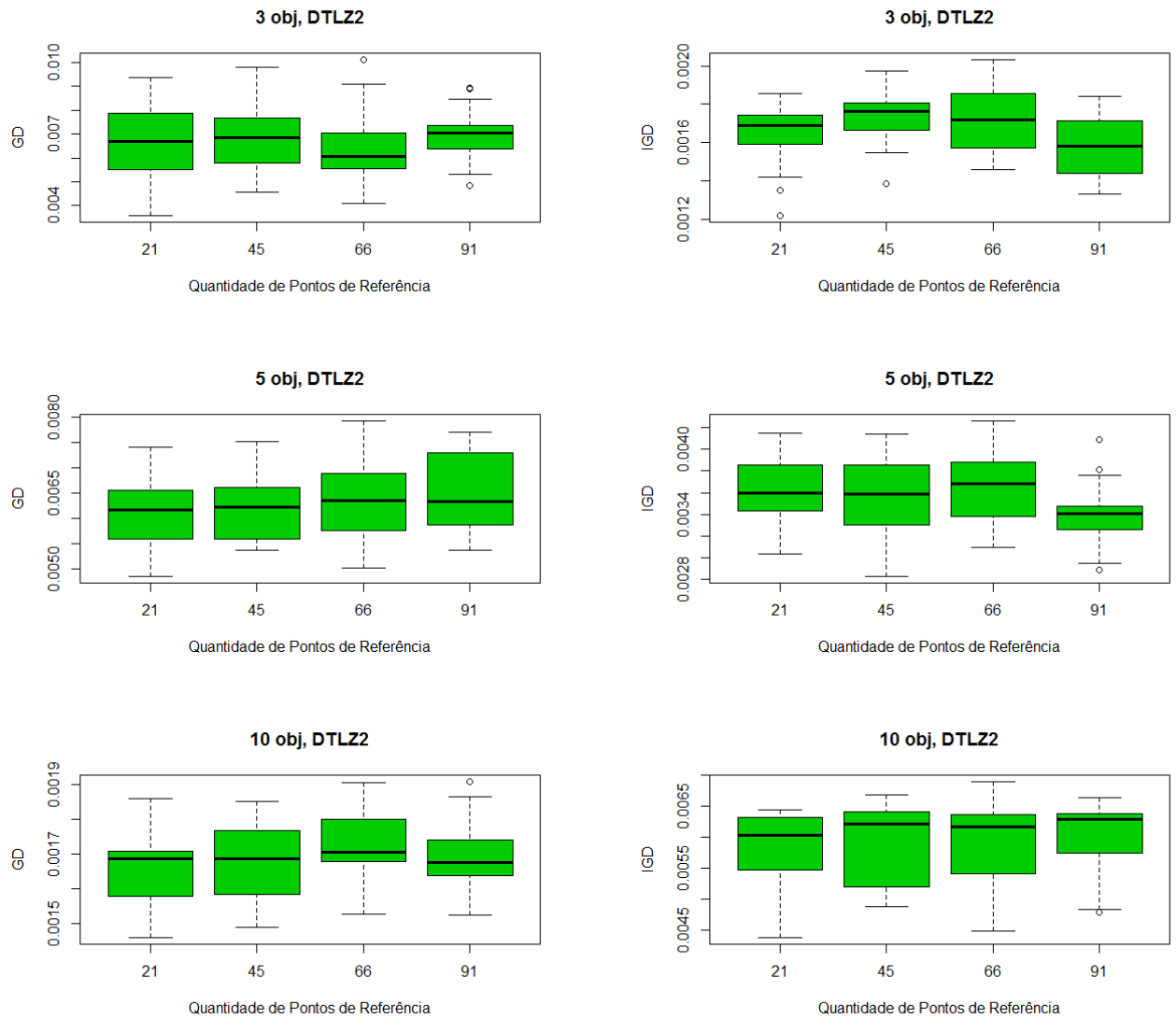


Figura 25 – Função côncava: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.

Para o problema do DTLZ2, as configurações que obtiveram bons resultados em termos de GD foram algoritmos com 21, 45 e 66 pontos conforme pode-se observar na Figura 25 e na Tabela 11. Enquanto em termos de IGD as configurações de 21 e 91 pontos se destacaram. O algoritmo com 21 pontos obteve os melhores resultados para o problema DTLZ2 no indicador GD e IGD quando aplicado a 10 funções objetivo, já a configuração de 91 pontos obteve melhores resultados em IGD para 3 e 5 objetivos.

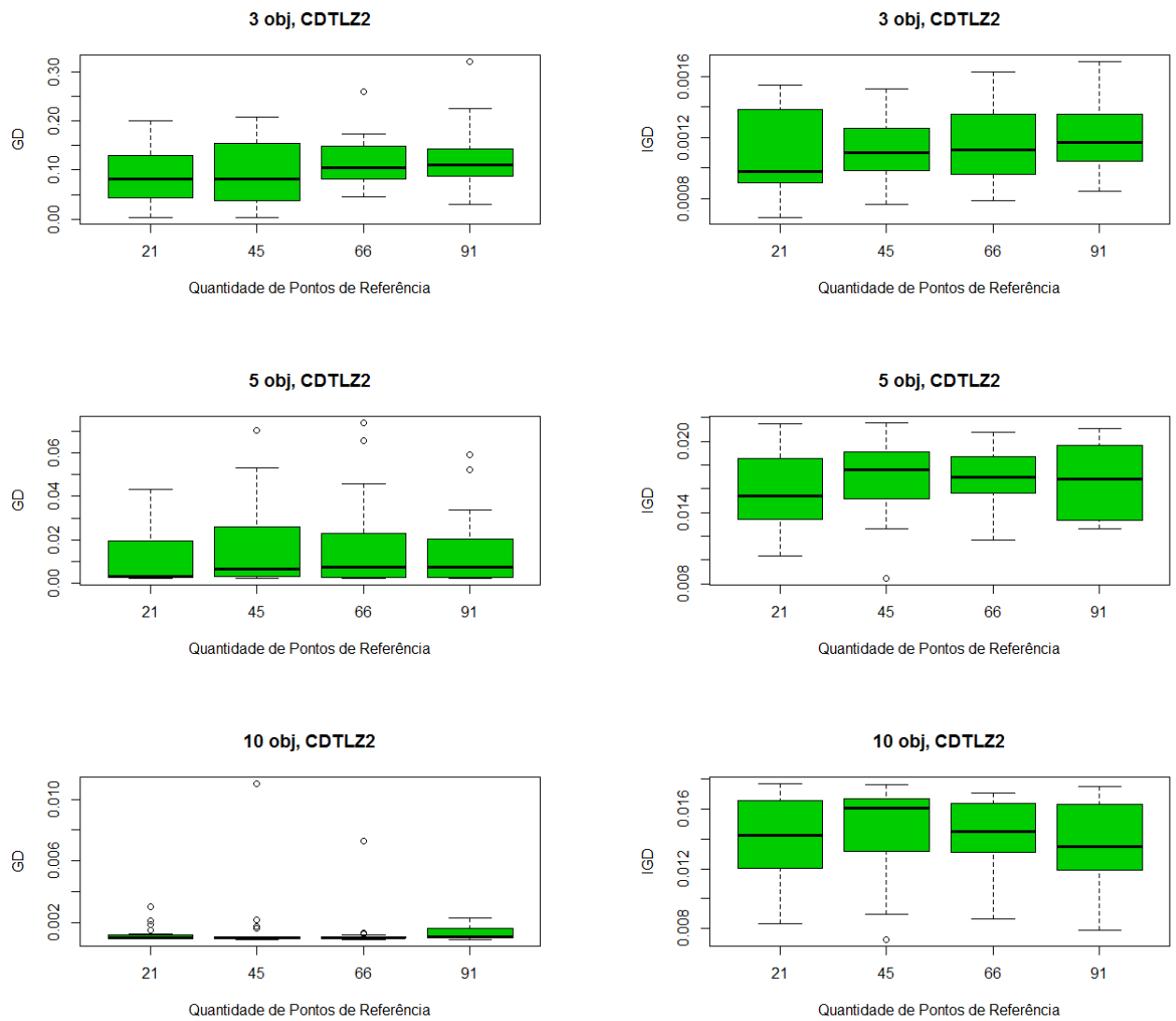


Figura 26 – Função côncava: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.

Para DTLZ2 Convexo, o algoritmo com 21 pontos pode ser destacado. Ele aparece com relevância em todas as funções objetivo no indicador GD, em IGD obteve mesmo êxito com 3 e 5 objetivos. O algoritmo com 91 pontos também conseguiu bons resultados, especialmente em termos de IGD para 10 objetivos como pode-se observar na Figura 26.

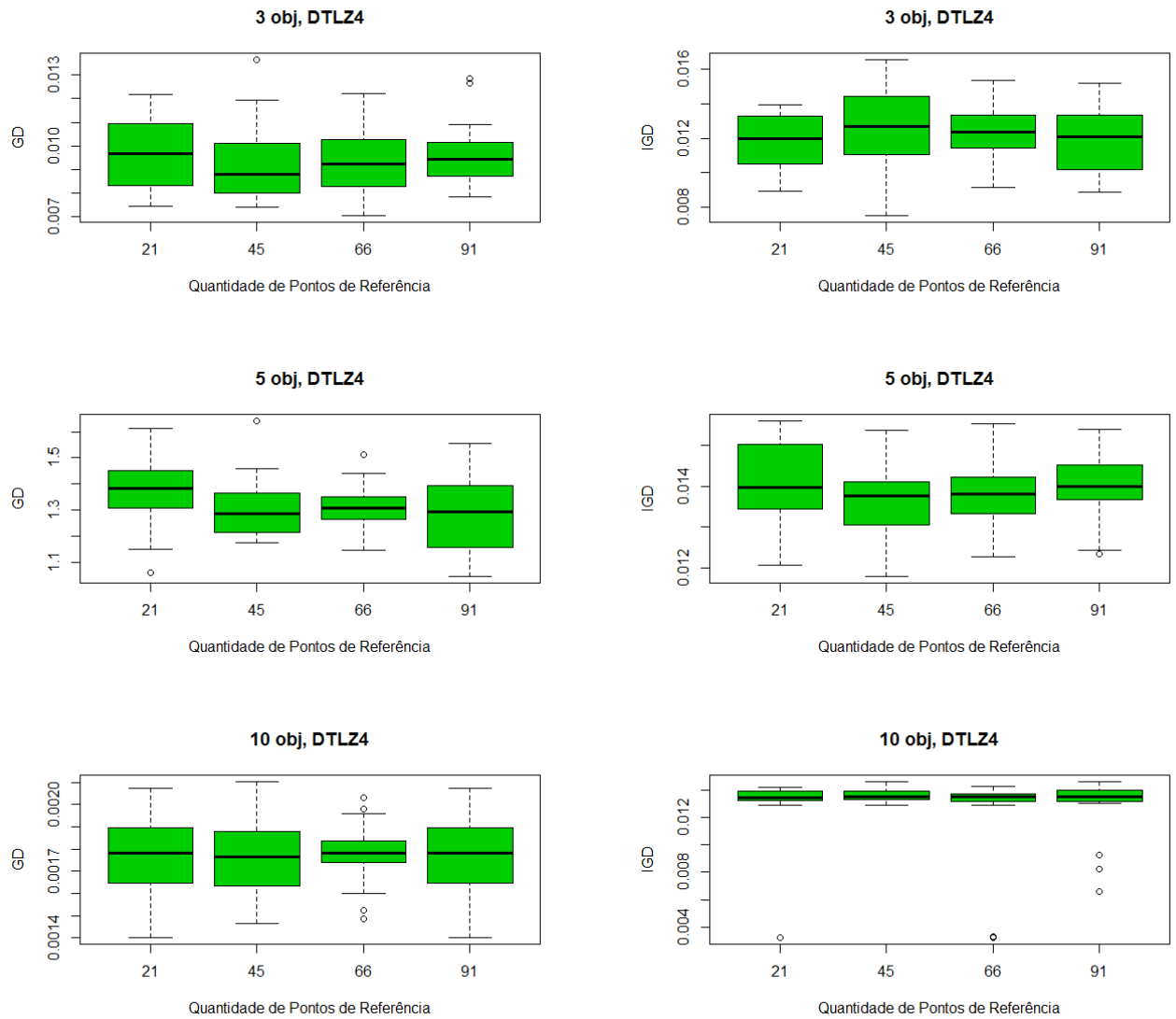


Figura 27 – Função côncava: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4

Como o problema DTLZ4 dificulta a obtenção de diversidade em frente a fronteira de Pareto e desejamos observar o comportamento dos algoritmos em termos de diversidade, a medida IGD é destacada como a medida mais relevante. O MSAT-MH que se destacou para maior dimensionalidade foi a configuração de 66 pontos de referência, pois conseguiu bons resultados no IGD para 10 objetivos, seguido por 21 e 45 pontos para 3 e 5 objetivos respectivamente, como pode-se observar na Figura 27 e na Tabela 12. Algoritmos com 45 e 91 pontos também tiveram resultados de GD significativos.

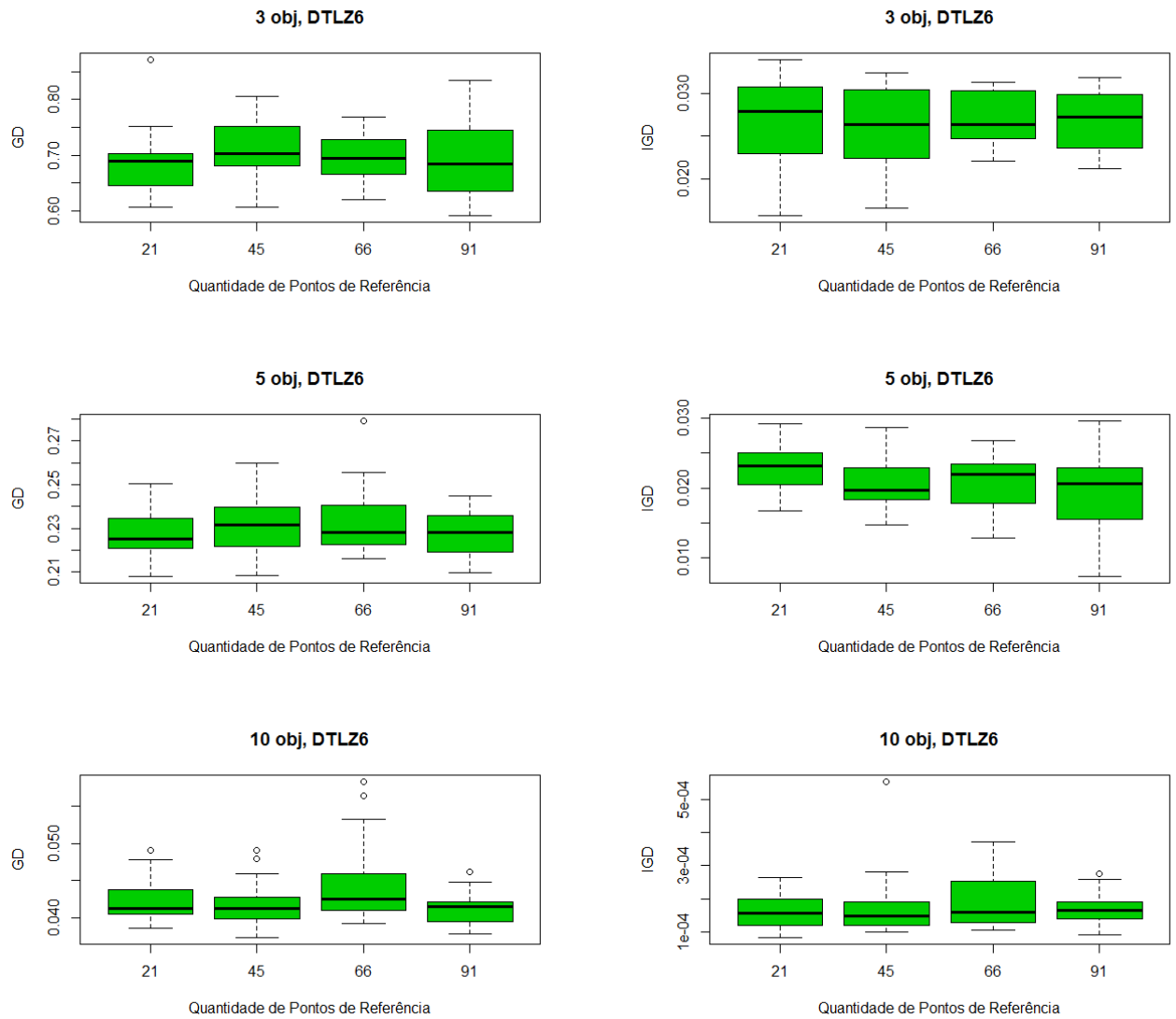


Figura 28 – Função côncava: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.

A medida GD é destacada como a mais relevante para o problema DTLZ6. Os algoritmos que tiveram valores relevantes em GD foram as configurações de 21 e 91 pontos de referência como mostra a Figura 28 e na Tabela 11. Além disso, os algoritmos com 21, 45 e 91 pontos possui os melhores valores de IGD.

Tabela 13 – Função convexa: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	6,8477E-03 (1,4787E-03)	6,2943E-03 (1,4230E-03)	6,8192E-03 (1,3456E-03)	7,0846E-03 (1,3671E-03)
	5	6,0209E-03 (4,8545E-04)	6,1227E-03 (7,1506E-04)	6,0034E-03 (8,0119E-04)	6,3722E-03 (9,5521E-04)
	10	1,6800E-03 (1,3200E-04)	1,7014E-03 (9,7901E-05)	1,6731E-03 (1,0764E-04)	1,7280E-03 (8,8637E-05)
DTLZ4	3	9,3931E-03 (1,1293E-03)	1,0044E-02 (1,5150E-03)	9,2219E-03 (1,2609E-03)	1,0074E-02 (1,3846E-03)
	5	1,3087E+00 (1,8007E-01)	1,2624E+00 (1,3238E-01)	1,3086E+00 (1,7801E-01)	1,3975E+00 (1,6662E-01)
	10	1,8298E-03 (1,2099E-04)	1,8444E-03 (1,5387E-04)	1,8405E-03 (1,9083E-04)	1,7843E-03 (1,4943E-04)
DTLZ6	3	7,1645E-01 (1,0442E-01)	6,6574E-01 (6,9019E-02)	7,0909E-01 (7,6783E-02)	7,2847E-01 (7,0561E-02)
	5	2,2971E-01 (1,3002E-02)	2,3149E-01 (1,2220E-02)	2,3255E-01 (8,3201E-03)	2,3318E-01 (9,3943E-03)
	10	4,2671E-02 (2,7244E-03)	4,1437E-02 (2,4651E-03)	4,1857E-02 (4,3308E-03)	4,2542E-02 (2,9872E-03)
DTLZ2 Convexo	3	1,3368E-01 (8,0605E-02)	9,2207E-02 (7,7777E-02)	1,2192E-03 (2,2592E-04)	1,0897E-01 (5,5479E-02)
	5	2,1551E-02 (2,5659E-02)	1,7319E-02 (1,8003E-02)	1,7537E-02 (2,4029E-02)	1,2904E-02 (1,3591E-02)
	10	1,8259E-03 (1,6801E-03)	1,3083E-03 (4,7406E-04)	1,9484E-03 (1,7882E-03)	1,6448E-03 (1,1994E-03)

Tabela 14 – Função convexa: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	1,6632E-03 (1,3709E-04)	1,6980E-03 (1,2167E-04)	1,6308E-03 (1,16807E-04)	1,6774E-03 (1,1623E-04)
	5	3,6176E-03 (3,4815E-04)	3,6527E-03 (2,9463E-04)	3,6532E-03 (3,0103E-04)	3,5405E-03 (3,4050E-04)
	10	5,8337E-03 (7,0886E-04)	5,8158E-03 (6,6701E-04)	5,9044E-03 (6,4422E-04)	6,0472E-03 (4,6716E-04)
DTLZ4	3	1,3012E-02 (1,4693E-03)	1,2619E-02 (2,1645E-03)	1,2731E-02 (2,0018E-03)	1,1521E-02 (1,1199E-03)
	5	1,3396E-02 (9,6620E-04)	1,3631E-02 (9,4296E-04)	1,4075E-02 (7,9665E-04)	1,3912E-02 (1,0742E-03)
	10	1,3262E-02 (1,7529E-03)	1,3061E-02 (2,2882E-03)	1,3145E-02 (2,110E-03)	1,2102E-02 (3,7749E-03)
DTLZ6	3	2,4847E-02 (4,3855E-03)	2,8410E-02 (4,0615E-03)	2,6476E-02 (3,5229E-03)	2,6662E-02 (2,8420E-03)
	5	2,1519E-02 (3,2273E-03)	1,9504E-02 (3,8538E-03)	2,1283E-02 (3,7438E-03)	2,0141E-02 (4,6029E-03)
	10	1,8551E-04 (7,3850E-05)	1,5634E-04 (5,3810E-05)	1,8558E-04 (1,0283E-04)	1,7761E-04 (6,2177E-05)
DTLZ2 Convexo	3	1,1892E-03 (2,4264E-04)	1,0558E-03 (2,7036E-04)	1,2192E-03 (2,2592E-04)	1,1365E-03 (1,8464E-04)
	5	1,6161E-02 (2,8871E-03)	1,6331E-02 (2,3296E-03)	1,6676E-02 (3,0598E-03)	1,6386E-02 (2,6651E-03)
	10	1,3459E-02 (2,7869E-03)	1,3600E-02 (2,7193E-03)	1,3983E-02 (2,1114E-03)	1,2258E-02 (3,1597E-03)

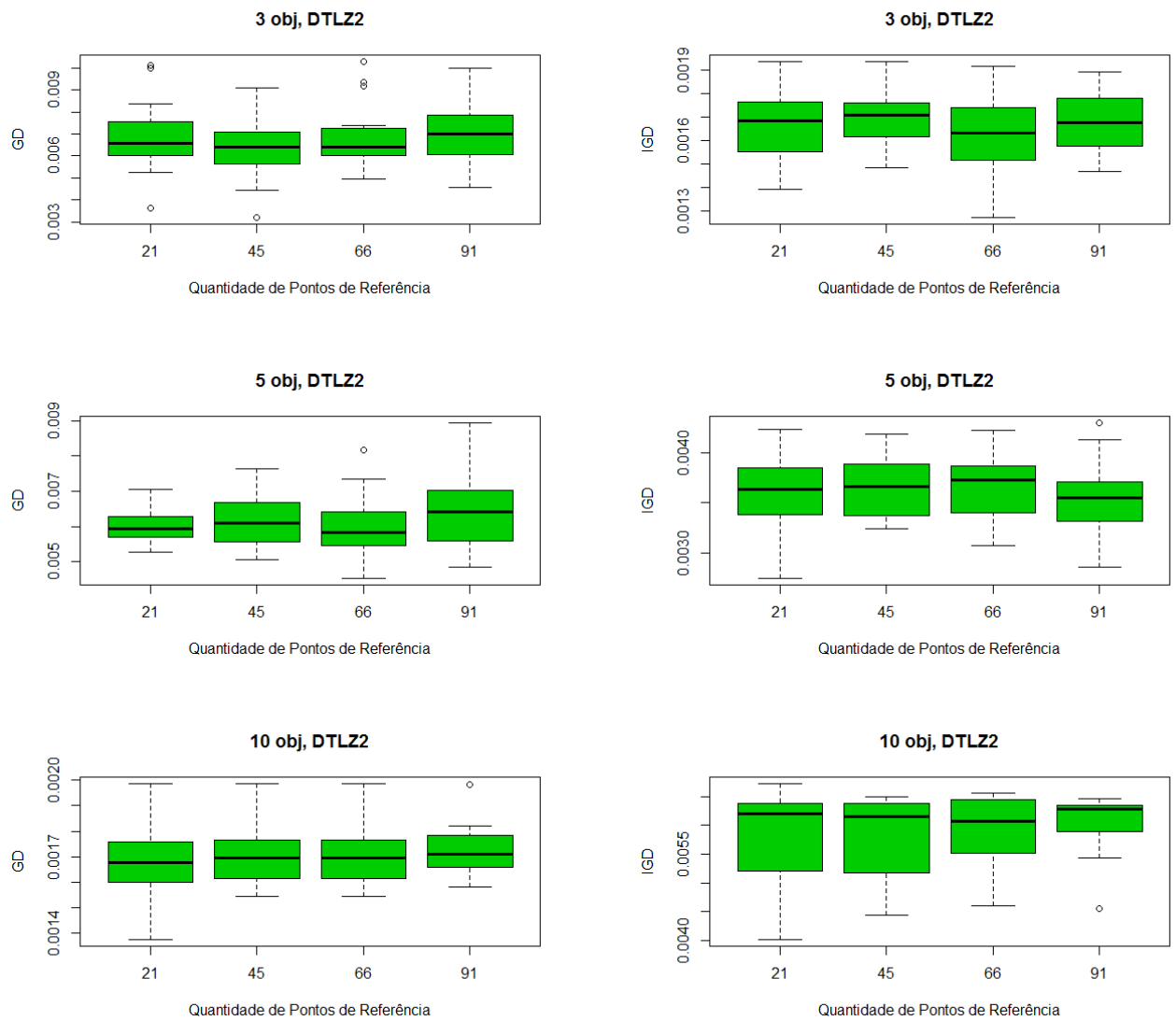


Figura 29 – Função convexa: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.

Para o problema do DTLZ2, as configurações que obtiveram bons resultados em termos de GD foram algoritmos com 45 e 66 pontos conforme pode-se observar na Figura 29 e na Tabela 13. O algoritmo com 66 pontos obteve os melhores resultados para o problema DTLZ2 no indicador GD quando aplicado a 5 e 10 funções objetivo. Enquanto em termos de IGD as configurações de 66, 91 e 45 pontos se destacaram para 3, 5 e 10 objetivos, respectivamente.

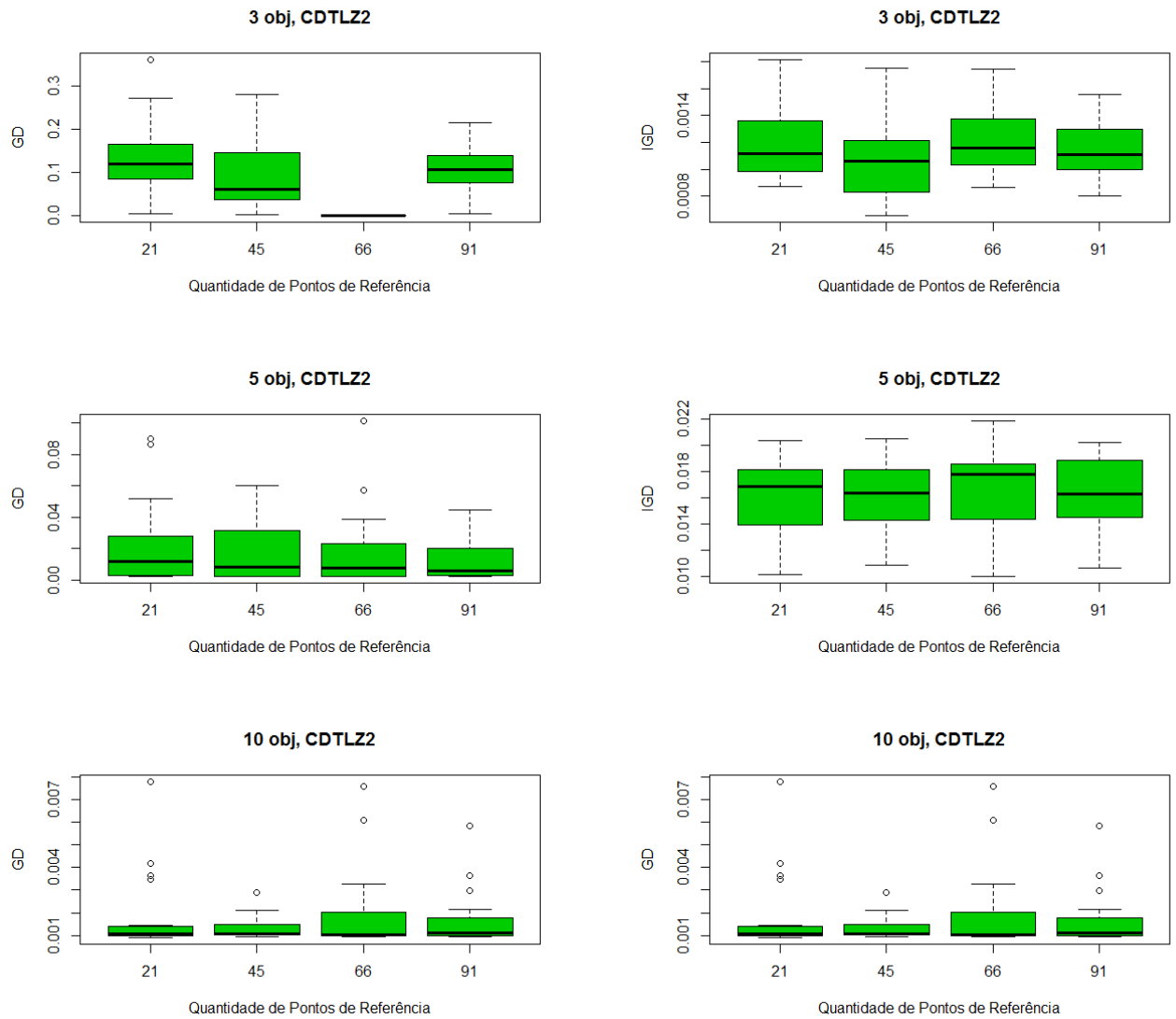


Figura 30 – Função convexa: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.

Os algoritmos que tiveram valores relevantes em GD foram as configurações de 45 pontos de referência para 10 objetivos, 66 para 3 objetivos e 91 para 5 objetivos como mostra a Figura 30. Além disso, os algoritmos com 21, 45 e 91 pontos possui os melhores valores de IGD para 5, 3 e 10 objetivos, respectivamente.

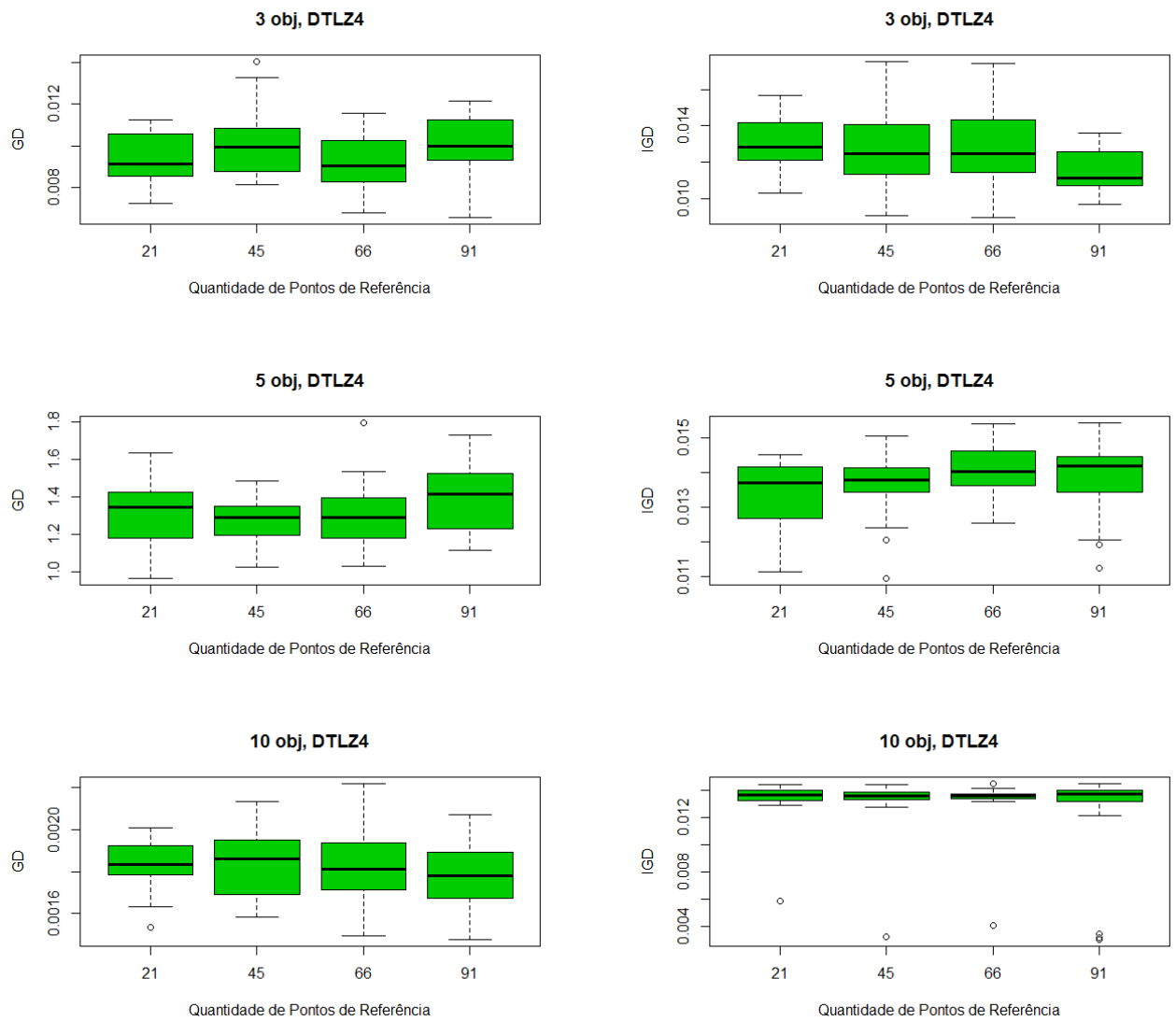


Figura 31 – Função convexa: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4

No problema DTLZ4 é aumentada a dificuldade na obtenção de diversidade em frente a fronteira de Pareto, com isso, a medida IGD é destacada como a medida mais relevante. O MSAT-MH que se destacou para maior dimensionalidade foi a configuração de 91 pontos de referência, pois conseguiu bons resultados para 3 e 10 objetivos no indicador IGD como pode-se observar na Figura 31. A configuração com 45 pontos se destacou para 5 objetivos, 66 para 3 objetivos e 91 para 10 objetivos no indicador GD.

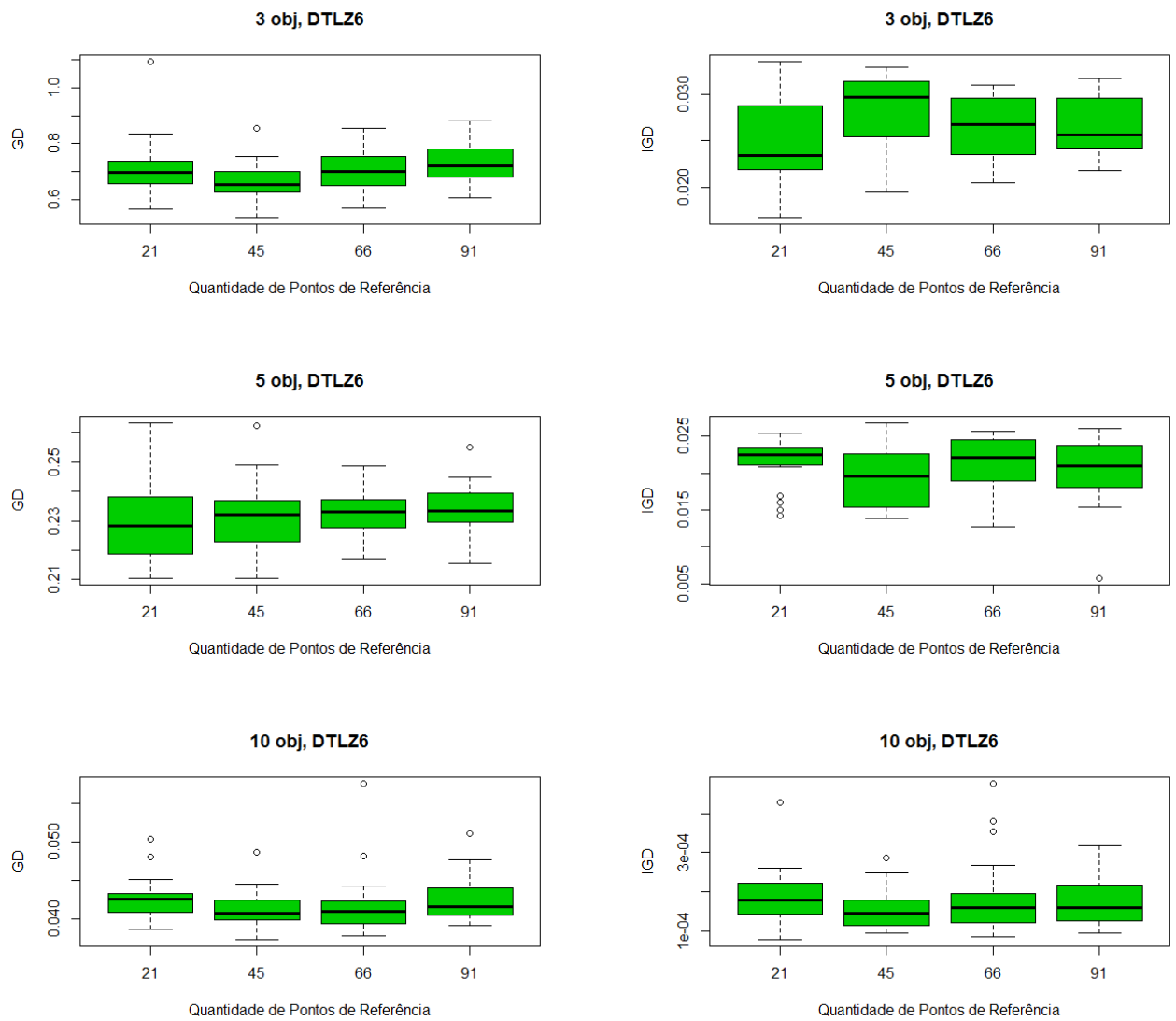


Figura 32 – Função convexa: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.

Para DTLZ6, o algoritmo com 45 pontos pode ser destacado. Ele aparece com relevância em quase todas as funções objetivo para ambos indicadores como pode-se observar na Figura 32 e nas Tabelas 13 e 14. O algoritmo com 21 pontos também conseguiu bons resultados em GD para 5 objetivos e em IGD para 3 objetivos.

Tabela 15 – Função senoidal: Valores médios e desvio padrão de GD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	6,8372E-03 (1,0297E-03)	6,4025E-03 (1,2629E-03)	6,3758E-03 (1,2386E-03)	6,4884E-03 (1,6455E-03)
	5	6,2968E-03 (9,1375E-04)	6,0727E-03 (5,2016E-04)	6,1787E-03 (4,9251E-04)	6,5366E-03 (1,3431E-03)
	10	1,6755E-03 (1,0946E-04)	1,6282E-03 (9,1738E-05)	1,7186E-03 (1,4251E-04)	1,6631E-03 (8,4122E-05)
DTLZ4	3	9,8839E-03 (1,3074E-03)	9,6326E-03 (1,7124E-03)	9,2917E-03 (1,0221E-03)	1,0034E-02 (1,8825E-03)
	5	1,2727E+00 (1,1350E-01)	1,3103E+00 (1,6459E-01)	1,3324E+00 (1,1953E-01)	1,2024E+00 (1,4584E-01)
	10	1,7672E-03 (1,9774E-04)	1,7351E-03 (1,5570E-04)	1,8667E-03 (1,4484E-04)	1,8207E-03 (1,6113E-04)
DTLZ6	3	7,0691E-01 (1,0554E-01)	7,1539E-01 (7,7314E-02)	7,0577E-01 (7,0792E-02)	6,8532E-01 (6,0586E-02)
	5	2,3067E-01 (8,7206E-03)	2,2844E-01 (1,3281E-02)	2,2891E-01 (1,4283E-02)	2,3278E-01 (1,1615E-02)
	10	4,1370E-02 (1,9791E-03)	4,1244E-02 (1,6491E-03)	4,2267E-02 (2,4971E-03)	4,3092E-02 (4,3542E-03)
DTLZ2 Convexo	3	8,9739E-02 (3,8705E-02)	6,9619E-02 (3,6493E-02)	9,4239E-02 (5,9848E-02)	1,0430E-01 (8,8870E-02)
	5	1,4000E-02 (1,7736E-02)	1,2487E-02 (9,8370E-03)	3,2067E-02 (3,5506E-02)	1,6018E-02 (1,9308E-02)
	10	1,8583E-03 (1,4192E-03)	1,8928E-03 (2,1792E-03)	1,8477E-03 (1,9935E-03)	1,7698E-03 (1,3126E-03)

Tabela 16 – Função senoidal: Valores médios e desvio padrão de IGD. Comparação principal (pós-teste de Kruskal-Wallis)

Problema	obj	21 pontos	45 pontos	66 pontos	91 pontos
DTLZ2	3	1,6201E-03 (1,6118E-04)	1,6198E-03 (1,4122E-04)	1,6862E-03 (1,3802E-04)	1,6722E-03 (1,6484E-04)
	5	3,5592E-03 (3,0218E-04)	3,6258E-03 (2,3066E-04)	3,4431E-03 (2,9131E-04)	3,5764E-03 (2,4292E-04)
	10	6,0380E-03 (5,2476E-04)	6,1380E-03 (5,5151E-04)	5,6943E-03 (7,7464E-04)	5,7642E-03 (7,0391E-04)
DTLZ4	3	1,2352E-02 (1,8887E-03)	1,2027E-02 (1,6819E-03)	1,2379E-02 (1,6083E-03)	1,2184E-02 (1,6510E-03)
	5	1,3570E-02 (9,0065E-04)	1,3663E-02 (8,9114E-04)	1,4243E-02 (6,6243E-04)	1,3986E-02 (4,8443E-04)
	10	1,3374E-02 (1,3471E-03)	1,2621E-02 (3,1689E-03)	1,2057E-02 (3,6132E-03)	1,1989E-02 (3,6553E-03)
DTLZ6	3	2,6885E-02 (5,2050E-03)	2,5731E-02 (5,0162E-03)	2,6786E-02 (3,4971E-03)	2,7356E-02 (4,9809E-03)
	5	2,0920E-02 (2,9146E-03)	2,0188E-02 (3,9449E-03)	2,1608E-02 (3,0620E-03)	2,0372E-02 (3,7169E-03)
	10	1,5377E-04 (6,1178E-05)	1,7068E-04 (5,0752E-05)	1,7730E-04 (7,3765E-05)	2,0307E-04 (1,7146E-04)
DTLZ2 Convexo	3	1,1201E-03 (1,9004E-04)	1,1273E-03 (1,7898E-04)	1,1539E-03 (2,1650E-04)	1,0821E-03 (1,6034E-04)
	5	1,7455E-02 (3,2453E-03)	1,5320E-02 (2,2656E-03)	1,6190E-02 (3,5220E-03)	1,6501E-02 (2,5873E-03)
	10	1,2818E-02 (2,6719E-03)	1,2659E-02 (2,9384E-03)	1,4562E-02 (2,5113E-03)	1,3423E-02 (3,2021E-03)

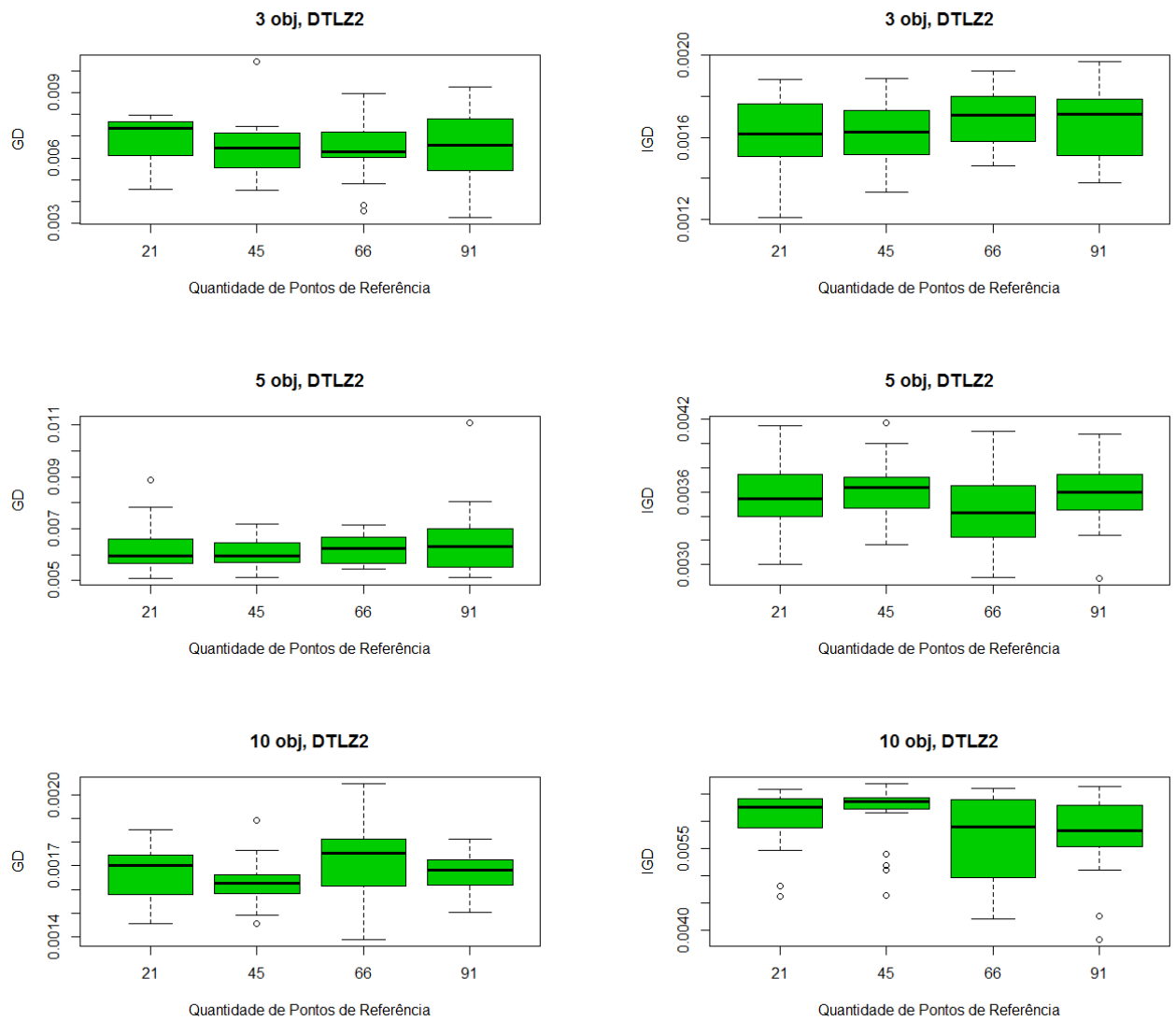


Figura 33 – Função senoidal: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2.

Para o problema do DTLZ2, as configurações que obtiveram bons resultados em termos de GD foram algoritmos com 45 e 66 pontos, conforme pode-se observar na Figura 33, no qual a configuração em com 45 pontos teve destaque para 5 e 10 objetivos, enquanto que a de 66 pontos destacou-se para 3 objetivos. Em termos de IGD o contrário aconteceu, onde a configuração de 66 obteve destaque para 5 e 10 objetivos e a de 45 pontos destacou-se para 3 objetivos.

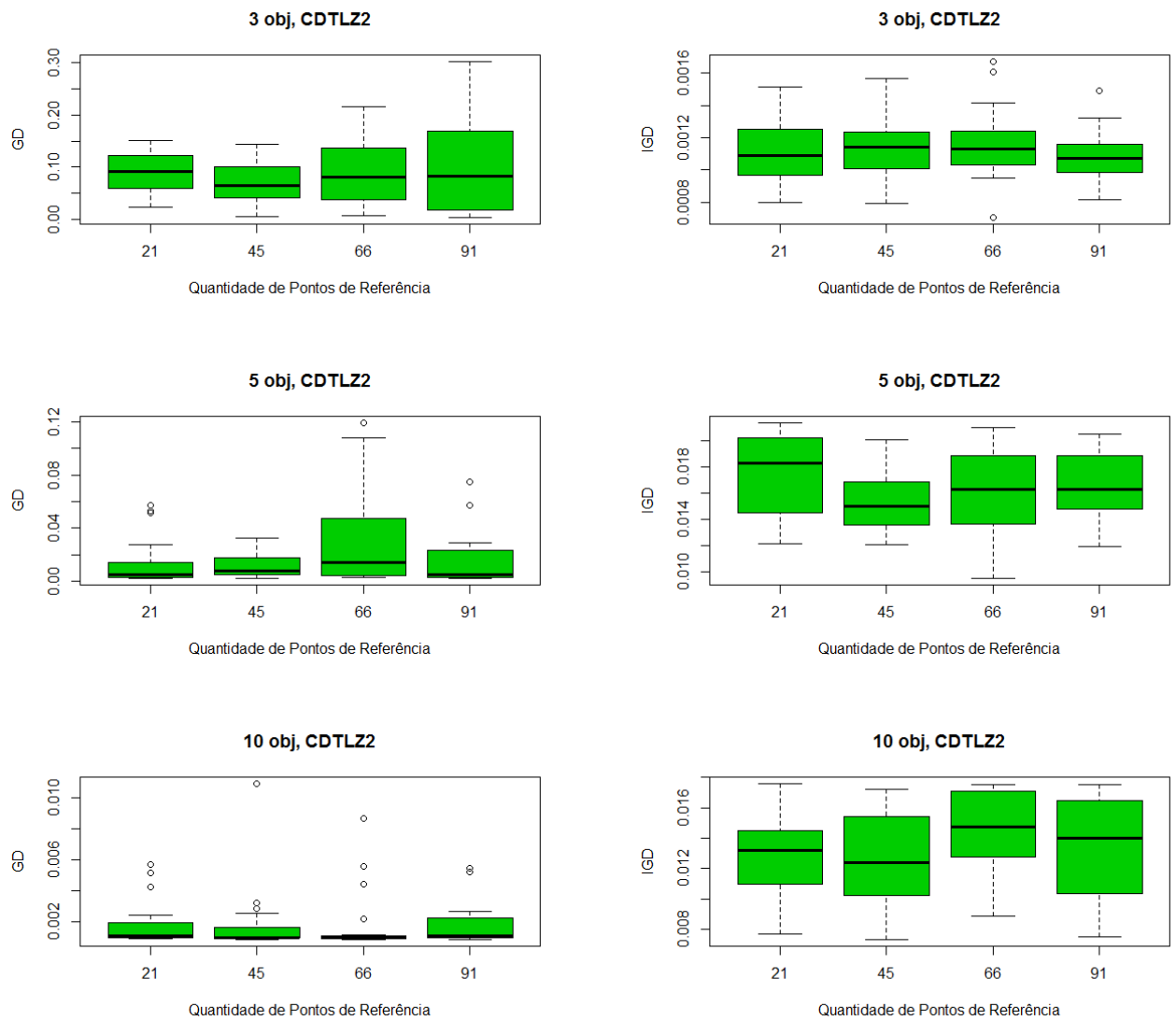


Figura 34 – Função senoidal: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ2 Convexo.

Para DTLZ2 Convexo, o algoritmo com 45 pontos pode ser destacado. Ele aparece com relevância para 5 e 10 objetivos no indicador IGD, além de 3 e 5 objetivos no indicador GD. O algoritmo com 91 pontos também conseguiu bons resultados, especialmente em termos de GD para 10 objetivos e em termos de IGD para 3 objetivos como pode-se observar na Figura 34.

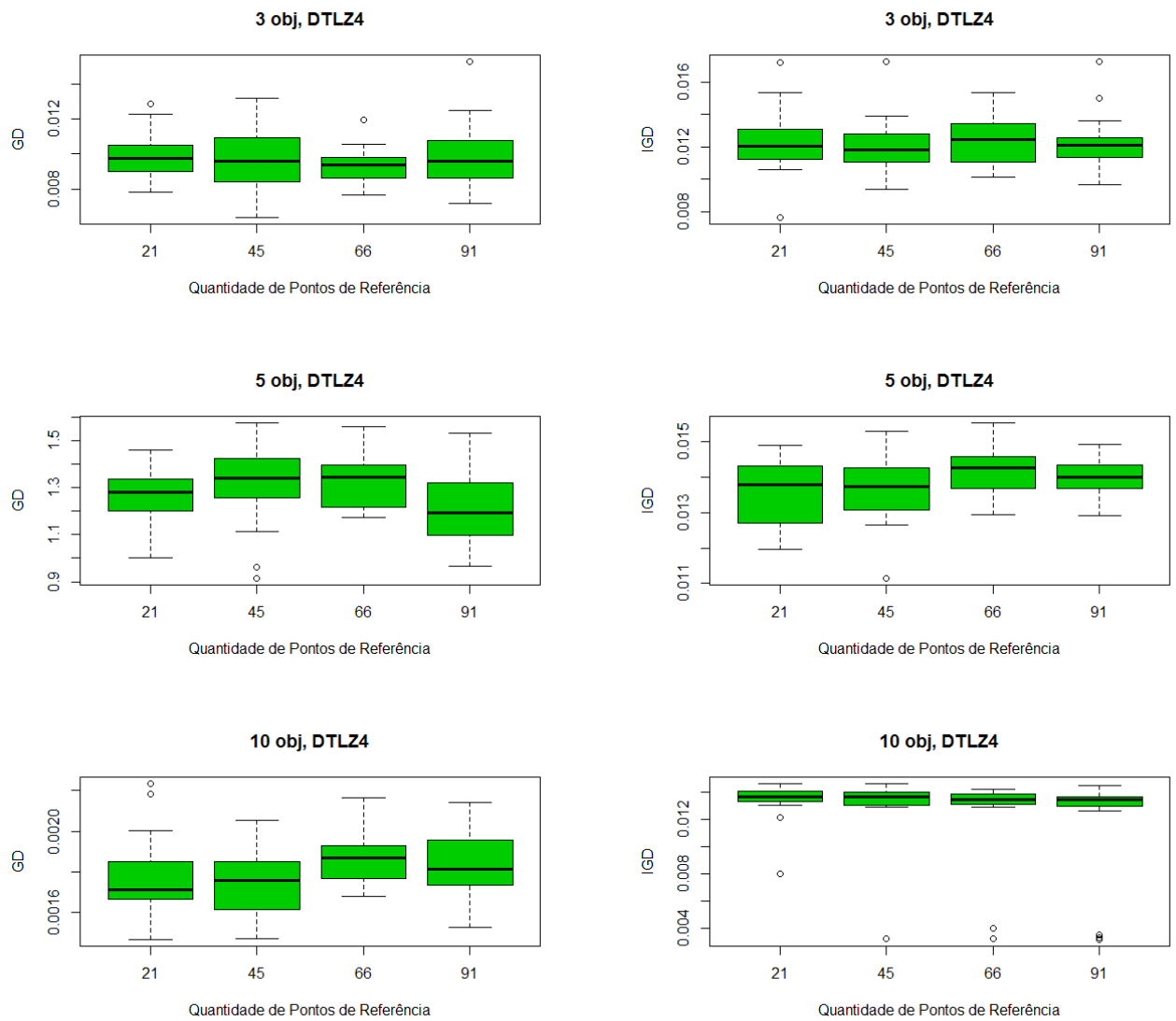


Figura 35 – Função senoidal: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ4

O problema DTLZ4 dificulta a obtenção de diversidade em frente a fronteira de Pareto, com isso a medida IGD é destacada como a medida mais relevante. Os MSAT-MH que se destacaram para 3, 5 e 10 objetivos, nessa ordem foram as configurações de 45, 21 e 91 pontos de referência, pois conseguiram bons resultados no IGD como pode-se observar na Figura 35. Algoritmos com 66, 91 e 45 pontos tiveram resultados de GD significativos para 3, 5 e 10 objetivos, respectivamente.

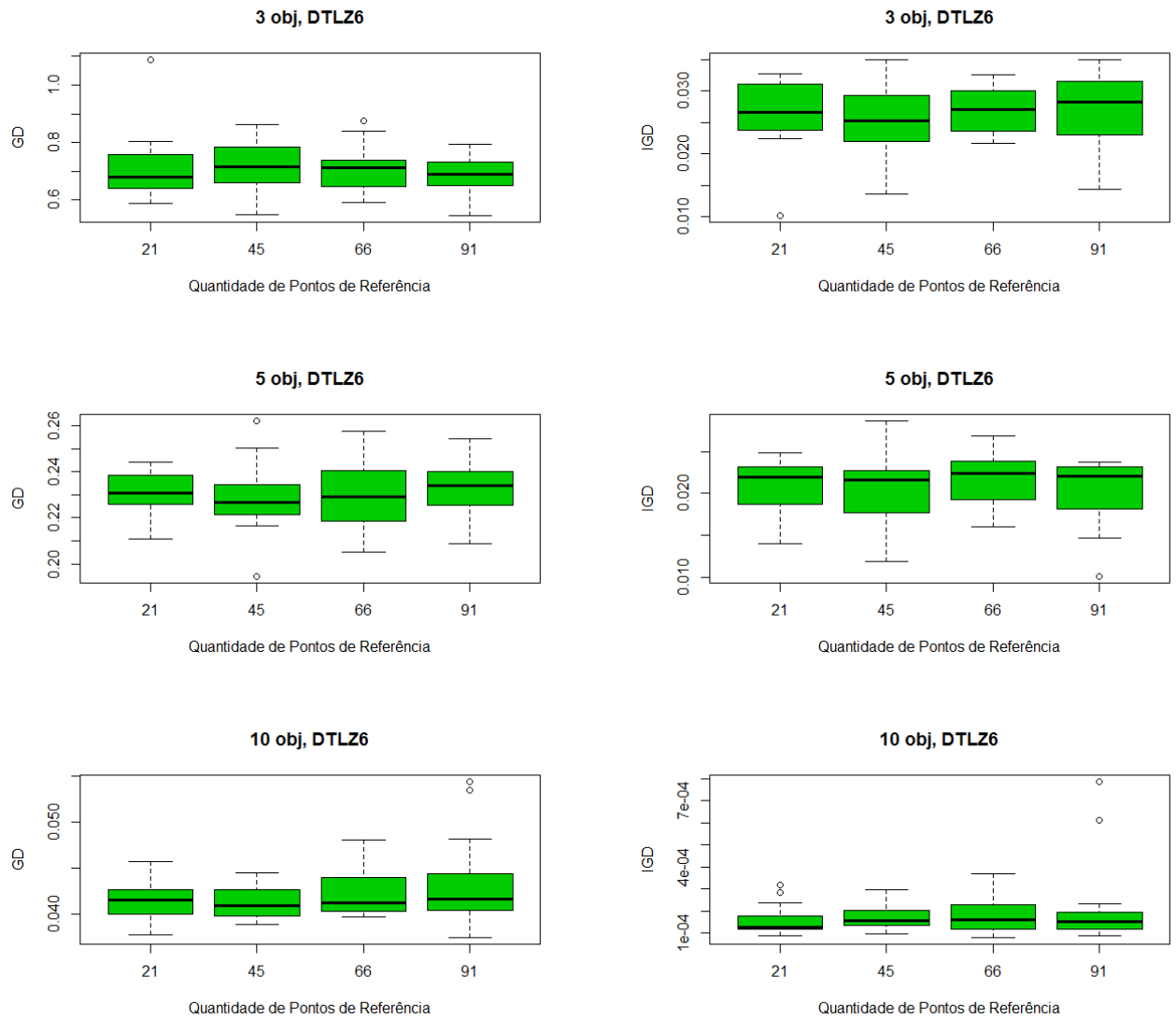


Figura 36 – Função senoidal: *Boxplot* dos valores GD (Esquerda) e IGD (Direita) com 3, 5 e 10 objetivos para o problema DTLZ6.

O problema DTLZ6 apresenta vários ótimos locais, isso torna a pesquisa mais difícil de convergir para a verdadeira fronteira de Pareto, com isso a medida GD é destacada como a mais relevante. O algoritmo que obteve valores relevantes em GD para 5 e 10 objetivos, foi o de 45 pontos de referência como mostra a Figura 36. Além disso, ele se destacou para 3 e 5 objetivos no indicador IGD.

Em resumo, o algoritmo com 21 pontos de referência geralmente apareceu em evidência na maioria dos problemas quando aplicado a função côncava. Já o algoritmo com 45 pontos de referência na maioria das vezes mostrou-se em evidência nos problemas quando aplicado a função convexa e a função senoidal. Ressaltando que, a configuração de 45 pontos quando aplicada a função senoidal o algoritmo apareceu mais vezes em evidência. Devido a esse resultado, foi escolhida essa configuração para ser comparada com o algoritmo MSAT, MSAT-H e NSGA-III.

4.7.2 Resultados e Discussão

Primeiro, as diferentes configurações do MSAT-MH foram confrontadas na seção anterior. As tabelas 17 e 18, diferente da análise anterior apresentam os valores médios e o desvio padrão de GD e IGD das melhores configurações de MSAT, MSAT-H e MSAT-MH que obteve os melhores resultados de acordo com o pós-teste de Kruskal-Wallis, para GD e IGD, respectivamente, em cada número de objetivos.

Tabela 17 – Comparação MSAT, MSAT-H, MSAT-MH e NSGA-III para o indicador GD.

Problema	obj	MSAT	MSAT-H	MSAT-MH	NSGAIII
DTLZ2	3	2,32E-04 (1,04E-05)	6,7811E-03 (1,6697E-03)	6,4025E-03 (1,2629E-03)	1,12E-03 (3,75E-05)
	5	1,31E-03 (6,36E-05)	2,7093E-02 (1,1032E-03)	6,0727E-03 (5,2016E-04)	2,52E-03 (1,97E-06)
	10	5,47E-03 (9,67E-05)	2,2935E-02 (3,3881E-04)	1,6282E-03 (9,1738E-05)	1,72E-02 (7,02E-03)
DTLZ4	3	1,35E-03 (5,62E-05)	9,4880E-03 (1,2102E-03)	9,6326E-03 (1,7124E-03)	1,11E-03 (2,50E-05)
	5	1,82E-01 (3,29E-03)	2,7617E-02 (3,7039E-03)	1,3103E+00 (1,6459E-01)	2,52E-03 (1,24E-06)
	10	3,66E-03 (1,32E-04)	2,2688E-02 (6,0464E-04)	1,7351E-03 (1,5570E-04)	3,34E-02 (1,19E-04)
DTLZ6	3	2,98E-06 (9,84E-08)	6,9963E-01 (5,4366E-02)	7,1539E-01 (7,7314E-02)	1,09E-02 (1,10E-02)
	5	3,32E-06 (1,44E-07)	7,7667E-02 (5,9768E-03)	2,2844E-01 (1,3281E-02)	4,27E-02 (3,13E-03)
	10	4,42E-06 (2,38E-07)	2,7896E-02 (1,0061E-03)	4,1244E-02 (1,6491E-03)	3,58E-01 (3,29E-01)
CDTLZ2	3	4,42E-04 (3,08E-05)	9,3580E-02 (7,0478E-02)	6,9619E-02 (3,6493E-02)	8,35E-04 (4,73E-05)
	5	6,04E-04 (2,46E-06)	1,4217E-02 (7,8692E-03)	1,2487E-02 (9,8370E-03)	8,19E-04 (1,01E-06)
	10	1,13E-03 (8,99E-06)	9,1940E-03 (2,1689E-04)	1,8928E-03 (2,1792E-03)	3,60E-03 (2,75E-04)

Tabela 18 – Comparação MSAT, MSAT-H, MSAT-MH e NSGA-III para o indicador IGD.

Problema	obj	MSAT	MSAT-H	MSAT-MH	NSGAIII
DTLZ2	3	1,77E-04 (4,00E-06)	1,6187E-03 (1,4928E-04)	1,6198E-03 (1,4122E-04)	5,94E-04 (1,25E-06)
	5	7,96E-04 (3,24E-05)	3,9709E-03 (5,4506E-04)	3,6258E-03 (2,3066E-04)	8,60E-04 (1,60E-06)
	10	1,64E-03 (7,27E-05)	9,1500E-03 (2,1208E-04)	6,1380E-03 (5,5151E-04)	3,00E-03 (9,52E-05)
DTLZ4	3	5,42E-03 (6,42E-04)	1,2649E-02 (1,6413E-03)	1,2027E-02 (1,6819E-03)	1,07E-03 (1,42E-03)
	5	7,38E-03 (3,87E-04)	1,7288E-02 (2,2625E-03)	1,3663E-02 (8,9114E-04)	8,59E-04 (6,69E-07)
	10	3,19E-04 (1,26E-05)	4,3552E-02 (2,7452E-03)	1,2621E-02 (3,1689E-03)	9,02E-04 (8,35E-06)
DTLZ6	3	7,49E-06 (4,56E-07)	2,6215E-02 (4,4512E-03)	2,5731E-02 (5,0162E-03)	2,96E-03 (3,66E-05)
	5	1,26E-05 (9,45E-07)	2,3829E-03 (3,0041E-04)	2,0188E-02 (3,9449E-03)	5,38E-03 (1,73E-04)
	10	1,84E-05 (1,92E-06)	8,4655E-03 (5,5346E-04)	1,7068E-04 (5,0752E-05)	2,05E-02 (4,21E-03)
CDTLZ2	3	1,77E-04 (3,71E-06)	1,0932E-03 (2,0286E-04)	1,1273E-03 (1,7898E-04)	6,98E-04 (1,41E-05)
	5	1,29E-03 (1,78E-05)	5,6711E-03 (1,3252E-03)	1,5320E-02 (2,2656E-03)	1,21E-03 (5,43E-06)
	10	1,62E-03 (6,12E-05)	8,4026E-03 (1,0726E-04)	1,2659E-02 (2,9384E-03)	3,54E-03 (8,31E-04)

Para DTLZ2, o MSAT-MH superou o MSAT o MSAT-H e o NSGA-III, em relação ao indicador GD com 10 funções objetivos, mas em termos de IGD não conseguiu superar o MSAT. O MSAT conseguiu ser melhor em todos os outros cenários, tanto em GD quanto em IGD para esse problema, se destacando na maioria das situações e por isso controlando melhor a deterioração na comparação com os outros algoritmos. DTLZ2 Convexo foi outro problema em que o MSAT superou o MSAT-H, MSAT-MH e o NSGA-III. No entanto, em problemas com menor dimensionalidade, com 5 funções objetivo, o NSGA-III teve os melhores resultados no indicador IGD. Os resultados mais inferiores de MSAT-H, MSAT-MH e MSAT sobre NSGA-III foram observados no DTLZ4. O NSGA-III obteve o melhor resultado em qualquer número de

objetivos e superou o MSAT-H em todas as configurações, perdendo apenas para o MSAT quando aplicado a 10 funções objetivo no indicador IGD e para o MSAT-MH para a mesma quantidade de objetivos no indicador GD. O DTLZ6 apresentou a maior diferença entre MSAT, MSAT-H, MSAT-MH e NSGA-III, onde os melhores resultados de MSAT sobre o MSAT-H, MSAT-MH e NSGA-III foram observados neste problema. O MSAT superou os outros para todas as medidas.

Em resumo, podemos concluir que a estratégia de variação de pontos de referência associado a aplicação de funções que modificam esses pontos melhoraram os resultados para os problemas DTLZ2 e DTLZ4 no cenário de 10 objetivos, porém para os outros problemas não obteve resultados satisfatórios quando comparados aos outros algoritmos. Outro algoritmo que se destacou para o problema DTLZ4 em ambos os indicadores foi o NSGA-III, só que para cenários com um baixo número de objetivos. O DTLZ6 apresentou a maior diferença entre MSAT, MSAT-H, MSAT-MH e NSGA-III, onde os melhores resultados de MSAT sobre os demais algoritmos foram observados neste problema. No geral, o MSAT superou todos os outros algoritmos para todas as medidas.

5

Considerações Finais

Nessa dissertação de mestrado foram desenvolvidos algoritmos MOPSO com *multi-swarms*, combinando métodos de arquivamento, topologias de comunicação e pontos de referência para resolver Problemas de Otimização com Muitos Objetivos. Foi escolhida a abordagem de *Multi-swarm*, pois, segundo os resultados obtidos na literatura é um bom caminho a seguir. Além disso, os algoritmos baseados no MOPSO se destacaram devido aos melhores resultados encontrados quando comparados à outras meta-heurísticas aplicadas aos MaOPs. Dessa forma, foram propostas novas estratégias com base em algoritmos MOPSO para resolver as dificuldades encontradas quando número de objetivos cresce.

Além de aprofundar os estudos da técnica de múltiplos enxames, foi adotado na dissertação a combinação de pontos de referência, topologias e arquivadores, já que também são abordagens que se destacam em alguns trabalhos da literatura no uso de Problemas com Muitos Objetivos. Com a combinação dessas abordagens, foram criados novos esquemas de múltiplos enxames e avaliado o desempenho dos algoritmos propostos em problemas de *benchmarking*, onde depois de verificar os resultados da abordagem, ficou claro que o trabalho teve um efeito positivo.

Através de uma extensa análise empírica, demonstrou-se que o MSAT é muito competitivo em comparação com o NSGA-III nos MaOPS. Primeiro, foi demonstrado que a topologia do anel combinada com os arquivadores Ideal e MGA é uma boa maneira de melhorar os resultados do MOPSO em otimização com muitos objetivos. Além disso, de acordo com a análise dos experimentos, o MSAT superou o NSGA-III em quase todos os cenários. O MSAT-H não destacou para nenhum objetivo, já MSAT-MH com estratégia de variação de pontos de referência e a aplicação de funções que modificam esses pontos melhoraram os resultados para os problemas DTLZ2 e DTLZ4 no cenário de 10 objetivos, porém para os outros problemas não obteve resultados satisfatórios.

Trabalhos futuros, desenvolvimento de um algoritmo que combine os diferentes conjuntos de pontos de referência num só algoritmo, adição do conjunto de teste WFG ([HUBAND et al., 2006](#)) e comparação do algoritmo proposto com outros algoritmos de otimização com muitos objetivos.

Referências

ADRA, S.; FLEMING, P. Diversity management in evolutionary many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, v. 15, n. 2, p. 183–195, abril 2011. Citado na página 45.

BRITTO, A.; MOSTAGHIM, S.; POZO, A. Iterated multi-swarm: a multi-swarm algorithm based on archiving methods. In: ACM. *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. [S.l.], 2013. p. 583–590. Citado na página 32.

BRITTO, A.; MOSTAGHIM, S.; POZO, A. Archive based multi-swarm algorithm for many-objective problems. In: IEEE. *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. [S.l.], 2014. p. 79–84. Citado 7 vezes nas páginas 13, 30, 31, 32, 34, 35 e 47.

BRITTO, A.; POZO, A. I-mopso: A suitable pso algorithm for many-objective optimization. In: IEEE. *Neural Networks (SBRN), 2012 Brazilian Symposium on*. [S.l.], 2012. p. 166–171. Citado 2 vezes nas páginas 32 e 47.

BRITTO, A.; POZO, A. Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization. In: IEEE. *Evolutionary Computation (CEC), 2012 IEEE Congress on*. [S.l.], 2012. p. 1–8. Citado 6 vezes nas páginas 13, 29, 30, 31, 33 e 47.

BRITTO, A.; POZO, A. Using reference points to update the archive of mopso algorithms in many-objective optimization. *Neurocomputing*, Elsevier, v. 127, p. 78–87, 2014. Citado 4 vezes nas páginas 13, 30, 32 e 34.

BRITTO, A.; POZO, A. Reference-point based multi-swarm algorithm for many-objective problems. In: IEEE. *Intelligent Systems (BRACIS), 2015 Brazilian Conference on*. [S.l.], 2015. Citado 5 vezes nas páginas 13, 30, 32, 34 e 35.

BROCKHOFF, D.; ZITZLER, E. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In: *Parallel Problem Solving from Nature-PPSN IX*. [S.l.]: Springer, 2006. p. 533–542. Citado na página 29.

CAMPOS, A. de; POZO, A. T.; DUARTE, E. P. A strategy to solve maops with multiple swarms based on broadcast communication. In: IEEE. *Intelligent Systems (BRACIS), 2013 Brazilian Conference on*. [S.l.], 2013. p. 256–262. Citado 6 vezes nas páginas 13, 29, 30, 31, 34 e 35.

CARVALHO, A. B. d. Novas estratégias para otimização por nuvem de partículas aplicadas a problemas com muitos objetivos. 2013. Citado 4 vezes nas páginas 14, 16, 22 e 28.

CHENG, R. et al. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 20, n. 5, p. 773–791, 2016. Citado 3 vezes nas páginas 33, 40 e 41.

CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 6, n. 1, p. 58–73, 2002. Citado na página 22.

COELLO, C.; LAMONT, G.; VELDHUIZEN, D. van. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer US, 2007. (Genetic and Evolutionary Computation). ISBN 9780387367972. Disponível em: <<https://books.google.com.br/books?id=rXIuAMw3IGAC>>. Citado 5 vezes nas páginas 12, 16, 19, 25 e 45.

COELLO, C. A. C.; LECHUGA, M. S. Mopso: A proposal for multiple objective particle swarm optimization. In: IEEE. *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. [S.l.], 2002. v. 2, p. 1051–1056. Citado na página 25.

DEB, K. et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, Springer, v. 1917, p. 849–858, 2000. Citado 2 vezes nas páginas 20 e 21.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 18, n. 4, p. 577–601, 2014. Citado 10 vezes nas páginas 13, 27, 28, 33, 41, 44, 45, 46, 49 e 53.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, v. 18, n. 4, p. 577–601, Aug 2014. ISSN 1089-778X. Citado na página 38.

DEB, K.; SAXENA, D. K. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Kangal report*, v. 2005011, 2005. Citado na página 29.

DEB, K.; SUNDAR, J. Reference point based multi-objective optimization using evolutionary algorithms. In: ACM. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. [S.l.], 2006. p. 635–642. Citado na página 29.

DEB, K. et al. Scalable multi-objective optimization test problems. In: IEEE. *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. [S.l.], 2002. v. 1, p. 825–830. Citado 4 vezes nas páginas 30, 31, 32 e 45.

DURILLO, J. J.; NEBRO, A. J. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, Elsevier, v. 42, n. 10, p. 760–771, 2011. Citado 2 vezes nas páginas 14 e 46.

ELLIOTT, A. C.; HYNAN, L. S. A sas® macro implementation of a multiple comparison post hoc test for a kruskal–wallis analysis. *Computer methods and programs in biomedicine*, Elsevier, v. 102, n. 1, p. 75–80, 2011. Citado 2 vezes nas páginas 45 e 48.

FIELDSEND, J. E.; SINGH, S. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In: *UK Workshop on Computational Intelligence (UKCI'02)*. [S.l.: s.n.], 2002. Citado na página 25.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. *Machine learning*, Springer, v. 3, n. 2, p. 95–99, 1988. Citado 2 vezes nas páginas 12 e 19.

HUBAND, S. et al. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 10, n. 5, p. 477–506, 2006. Citado na página 85.

- ISHIBUCHI, H. et al. Behavior of emo algorithms on many-objective optimization problems with correlated objectives. In: IEEE. *Evolutionary Computation (CEC), 2011 IEEE Congress on.* [S.l.], 2011. p. 1465–1472. Citado na página 28.
- ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: CITESEER. *IEEE congress on evolutionary computation.* [S.l.], 2008. p. 2419–2426. Citado na página 28.
- JAIMES, A. L.; COELLO, C. A. C. Some techniques to deal with many-objective problems. In: ACM. *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Graduate Student Workshop.* [S.l.], 2009. p. 2693–2696. Citado na página 27.
- JAIMES, A. L.; COELLO, C. A. C. Study of preference relations in many-objective optimization. In: ACM. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation.* [S.l.], 2009. p. 611–618. Citado 2 vezes nas páginas 19 e 30.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Neural Networks, 1995. Proceedings., IEEE International Conference on.* [S.l.], 1995. v. 4, p. 1942–1948. Citado na página 12.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on.* [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4. Citado na página 22.
- KNOWLES, J.; CORNE, D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: IEEE. *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.* [S.l.], 1999. v. 1. Citado na página 21.
- LAUMANN, M.; ZENKLUSEN, R. Stochastic convergence of random search methods to fixed size pareto front approximations. *European Journal of Operational Research*, Elsevier, v. 213, n. 2, p. 414–421, 2011. Citado na página 47.
- LI, K. et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 19, n. 5, p. 694–716, 2015. Citado na página 33.
- LÓPEZ-IBÁÑEZ, M.; KNOWLES, J.; LAUMANN, M. On sequential online archiving of objective vectors. In: SPRINGER. *EMO.* [S.l.], 2011. p. 46–60. Citado na página 26.
- MATOS, J. L.; BRITTO, A. Multi-swarm algorithm based on archiving and topologies for many-objective optimization. In: IEEE. *Evolutionary Computation (CEC), 2017 IEEE Congress on.* [S.l.], 2017. p. 1877–1884. Citado na página 48.
- MOSTAGHIM, S.; TEICH, J. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In: IEEE. *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE.* [S.l.], 2003. p. 26–33. Citado 2 vezes nas páginas 25 e 32.
- NEBRO, A. J. et al. Smpso: A new pso-based metaheuristic for multi-objective optimization. In: IEEE. *Computational intelligence in multi-criteria decision-making, 2009. mcdm'09. iee symposium on.* [S.l.], 2009. p. 66–73. Citado 3 vezes nas páginas 27, 29 e 36.

- NEBRO, A. J. et al. A study of convergence speed in multi-objective metaheuristics. In: SPRINGER. *International Conference on Parallel Problem Solving from Nature*. [S.l.], 2008. p. 763–772. Citado na página 47.
- OLIVEIRA, M. C. de; CARVALHO, A. B. de. *Um Estudo Sobre a Influência dos Pontos de Referência na Otimização com Muitos Objetivos*. [S.l.], 2017. Citado 5 vezes nas páginas 6, 39, 40, 41 e 42.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm intelligence*, Springer, v. 1, n. 1, p. 33–57, 2007. Citado 2 vezes nas páginas 12 e 19.
- PULIDO, G. T. On the use of self-adaptation and elitism for multiobjective particle swarm optimization. 2005. Citado na página 24.
- REYES-SIERRA, M.; COELLO, C. C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, v. 2, n. 3, p. 287–308, 2006. Citado 8 vezes nas páginas 6, 12, 17, 19, 21, 23, 24 e 32.
- RUDOLPH, G. *Evolutionary search under partially ordered fitness sets*. [S.l.], 2001. Citado na página 20.
- SATO, H.; AGUIRRE, H. E.; TANAKA, K. Controlling dominance area of solutions and its impact on the performance of moeas. In: SPRINGER. *Evolutionary multi-criterion optimization*. [S.l.], 2007. p. 5–20. Citado 3 vezes nas páginas 29, 30 e 33.
- SCHÜTZE, O.; LARA, A.; COELLO, C. A. C. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Trans. Evolutionary Computation*, v. 15, n. 4, p. 444–455, 2011. Citado 2 vezes nas páginas 28 e 45.
- TEAM, R. C. et al. R: A language and environment for statistical computing. Vienna, Austria, 2013. Citado na página 46.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 11, n. 6, p. 712–731, 2007. Citado na página 29.
- ZHANG, Q.; LIU, W.; LI, H. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: *IEEE congress on evolutionary computation*. [S.l.: s.n.], 2009. v. 1, p. 203–208. Citado na página 32.
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, MIT Press, v. 8, n. 2, p. 173–195, 2000. Citado na página 20.
- ZITZLER, E.; KÜNZLI, S. Indicator-based selection in multiobjective search. In: SPRINGER. *Parallel Problem Solving from Nature-PPSN VIII*. [S.l.], 2004. p. 832–842. Citado na página 29.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: GIANNAKOGLU, K. et al. (Ed.). *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*. [S.l.]: International Center for Numerical Methods in Engineering (CIMNE), 2002. p. 95–100. Citado na página 47.

ZITZLER, E. et al. *SPEA2: Improving the strength Pareto evolutionary algorithm*. [S.l.]: Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), 2001. Citado na página 25.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: SPRINGER. *Parallel problem solving from nature—PPSN V*. [S.l.], 1998. p. 292–301. Citado 2 vezes nas páginas 29 e 30.